P.S.I.

INTELLIGENCE ARTIFICIELLE

I.A. SUR AMSTRAD CPC
LANGAGE ET FORMES



THIERRY ET ÉRIC
LÉVY-ABÉGNOLI

I.A. SUR AMSTRAD CPC
LANGAGE ET FORMES

ISBN: 2-86595-325-4

INTELLIGENCE ARTIFICIELLE

I.A. SUR AMSTRAD CPC
LANGAGE ET FORMES

THIERRY ET ÉRIC
LÉVY-ABÉGNOLI



ÉDITIONS DU P.S.I.
1986

Eric Lévy-Abégnoli

Ingénieur de l'école centrale de Lyon, il travaille actuellement au laboratoire de recherche et développement chez un grand constructeur. Sa formation et son métier l'ont amené à acquérir des compétences pratiques dans plusieurs domaines ayant trait à l'intelligence artificielle. A son actif, citons un logiciel d'aide au diagnostic vétérinaire, un logiciel permettant la lecture et l'interprétation de radiographies, un système de gestion de bases de données relationnelles. Ajoutons à celà plusieurs publications dont une série d'articles sur les bases de données.

Thierry Lévy-Abégnoli

De formation scientifique, il est journaliste indépendant dans la presse informatique et auteur de logiciels. Sa principale réalisation est un système expert commercialisé dont le moteur d'inférence est d'ordre un. Egalement spécialiste de la programmation des jeux de stratégie, il est l'auteur d'une série d'articles traitant de ce sujet dans l'Ordinateur Individuel et a terminé sixième de l'édition 83 du tournoi international d'Othello sur ordinateur. Son domaine couvre également la programmation des jeux d'arcades en assembleur.

Achevé d'imprimer
sur les presses de l'imprimerie IBP
à Rungis (Val-de-Marne 94) (1) 46.86.73.54
Dépôt légal - Septembre 1986

N° d'impression: 4889
N° d'édition: 86595-325-1
N° d'ISBN: 2-86595-325-4

DISQUETTE D'ACCOMPAGNEMENT

CATALOGUE DE LA DISQUETTE D'ACCOMPAGNEMENT

Drive A: user 0

ALPHAB		12K	ROBOT	.BAS	15K
BALANCE	.BAS	11K	SEMAN	.BAS	14K
EXPHRA	.	1K	SOPHIE	.BAS	11K
P4	.BAS	10K	SOURIS	.BAS	8K
PHRASES	.BAS	8K	SYNTAX	.BAS	4K
RECFORM	.BAS	12K	TTT	.BAS	9K

Cette disquette est notamment destinée à tous ceux qui n'ont pas le temps de recopier les programmes de ce livre. La disquette contient, d'une part les 10 logiciels figurant dans cet ouvrage (suffixe .BAS), et d'autre part deux fichiers servant d'exemple :

– EXPHRA est une petite base de connaissances destinée au logiciel PHRASES ;
– ALPHAB est une base d'objets destinée au logiciel RECFORM. Ces objets représentent les 26 lettres de l'alphabet.

A noter que ces deux fichiers sont facilement exploitables en utilisant les options de chargement des logiciels concernés.

SOMMAIRE

INTRODUCTION

INTELLIGENCE ARTIFICIELLE : OÙ EN SOMMES-NOUS ?

N'est-il pas étonnant de constater le fabuleux pouvoir évocateur de certains mots. Ainsi, quelle richesse, quelle palette de sentiments, de doutes, d'images fortes et subtiles dans celui-là : Intelligence. Prétendre donner une définition, aussi rigoureuse ou tortueuse soit-elle, n'est-ce pas une gageure ? Une définition, pour un tel mot, ne peut prétendre qu'au rang d'ersatz tant elle vide le mot de son contenu émotionnel. Et puis, quelle vanité ! L'intelligence est un domaine si complexe, si vaste et si peu connu qu'il faudrait soi-même en posséder une, hors du commun, pour en expliquer la nature. Que dire alors de l'intelligence artificielle (IA), qui voudrait copier son modèle humain : pour copier, il faut connaître parfaitement l'original, avoir tout analysé, tout compris et tout expliqué.

Et pourtant l'intelligence artificielle existe, je l'ai rencontrée... comme vous la rencontrerez aussi au fil de ces pages. Bien sûr, l'intelligence artificielle n'en est qu'à ses balbutiements. Bien sûr, le chemin qui lui reste à parcourir pour égaler son illustre modèle est sans commune mesure avec celui déjà effectué. L'intelligence artificielle n'est qu'un bambin bruyant et trop fier de ses premières dents de lait : il lui faudra de nombreuses années, peut-être des siècles pour atteindre l'âge adulte. Mais tout de même, des réussites spectaculaires jalonnent déjà son histoire si courte. Qui s'étonnerait de ces titres dans un journal télévisé :

– "les Etats-Unis réalisent pour la première fois l'interception en vol d'un missile grâce à un petit engin "intelligent" capable d'adapter sa trajectoire aux conditions atmosphériques et aux caprices de sa cible.

– un grand maître d'Echecs connaît une défaite cinglante : son adversaire était un ordinateur.

– un bateau très particulier a levé l'ancre aujourd'hui : un ordinateur tiendra lieu à bord de seule compétence médicale.

– des enfants sourds de naissance pourront entendre... par les yeux ! Un micro-ordinateur rend désormais cela possible en proposant une représentation visuelle des sons..."

Science-fiction ? Ne croyez pas cela. Les systèmes informatiques de type automate ou robot sont désormais capables de s'adapter aux variations d'environ-

nement. Les programmes d'Echecs sont à moins de 500 points d'ELD du champion du monde. Le système expert baptisé "Bateau Sans Médecin" (BSM), développé en 1982 dans le centre scientifique IBM de Paris permet de simuler la prise de décision médicale à bord d'un bateau. Toujours dans le domaine médical, d'autres systèmes experts permettent le pronostic de certains cancers, et MYCIN peut faire des diagnostics sur les infections du sang et les méningites, dont la qualité égale celle atteinte par les meilleurs spécialistes. Un prototype de système de visualisation de la voix, développé autour d'un IBM-PC est utilisé depuis 1979 par l'Institut National des Jeunes Sourds de Paris (INJS).

Les exploits de ces petites machines, les ordinateurs, et surtout des programmes et autres systèmes experts qui les animent, font désormais partie de notre vie quotidienne. Dotées d'une faible parcelle d'intelligence, elles parviennent même parfois à surpasser l'homme dans des domaines spécifiques.

DU CALME, DU CALME...

Il convient d'être réaliste et de replacer dans leur contexte ces quelques réussites. Il ne saurait être question de renouveler les erreurs passées, à une époque pas si lointaine où beaucoup prétendaient ériger la technologie au rang de divinité omnipotente. La technologie a ses limites que l'on connaît bien désormais, instruits que nous sommes par quelques accidents célèbres et quelques échecs cinglants. Il en est de même de l'informatique et autres sciences plus portées sur le verbe que sur la matière.

L'homme dispose de cette capacité originale de traiter des problèmes de nature très différente, depuis les jeux de stratégie jusqu'à la prédiction du temps. Plus sérieusement, le test de Turing permet d'évaluer l'intelligence d'un système expert ou de toute autre machine prétendant à l'intelligence. Un opérateur a devant lui deux terminaux, le premier contrôlé par un homme en coulisses et le second par l'intelligence artificielle. Il pose des questions à partir des deux terminaux et selon les réponses, doit pouvoir distinguer l'homme de la machine. S'il n'y parvient pas, l'intelligence artificielle est jugée l'égale de l'homme : vous l'aviez compris, la distinction est (et pour encore longtemps) très aisée à faire. L'homme a encore de beaux jours devant lui !

Réservons donc la futurologie pour la conclusion de cet ouvrage et faisons le point sur ce que peuvent faire les machines d'aujourd'hui. A la question : existe-t-il une machine dotée d'intelligence artificielle et capable de rivaliser dans tous les domaines avec un homme ordinaire, les spécialistes répondent : non !

A la question : dans combien de domaines spécifiques de l'intelligence l'homme est-il supérieur à la machine, ces mêmes spécialistes répondent "très peu" ! Cette dernière réflexion justifie à elle seule l'étonnement, voire l'émerveillement d'un public non averti. Car cette vérité est tout de même dure à avaler : il n'existe presque aucun domaine où il ne soit possible de concevoir une intelligence artificielle pouvant rivaliser avec un homme ordinaire. Et pourtant, les chercheurs et tous les spécialistes en informatique, robotique, intelligence artificielle, ainsi que les linguistes et même les philosophes le savent bien : ce qui a été fait en matière d'intelligence artificielle, ou ce qui ne l'a pas été mais qui appartient au domaine du possible, constitue un tout petit pas vers un but très éloigné. Et pour atteindre ce but, que l'on pourrait figurer par l'androïde intelligent décrit dans les romans d'Isaac Asimov, mille ans et mille milliards de dollars n'y suffiront sans doute pas.

Alors, du calme, du calme... Gardons les pieds sur terre, analysons ce premier pas, apportons-y notre modeste contribution (10 programmes couvrant tous les domaines de l'intelligence artificielle) et tentons d'imaginer quelle sera la prochaine étape dans cette aventure passionnante que constitue l'intelligence artificielle.

AU FAIT DE QUOI PARLONS-NOUS ?

Il n'est pas de domaine plus complexe que l'intelligence, de régions aussi peu explorées que celles qui s'étendent au-delà de notre boîte crânienne. Définir en quelques mots ou quelques phrases l'intelligence constitue un exercice bien périlleux. Beaucoup s'y sont cassé les dents et les rigoristes plutôt davantage que les amoureux de la périphrase.

Laissons de côté les définitions de type métaphoriques qui, quoique non dénuées d'intérêt, n'apportent pas grand chose au débat : "c'est ce qui distingue l'homme de la bête", et qui ouvrent la porte aux pires boutades : "c'est ce qui distingue la bête de l'homme".

Attardons-nous un instant sur celle-ci : "Aptitude à s'adapter à une situation, faculté de comprendre..." (Larousse). Cette définition (incomplète car elle ne laisse rien entrevoir des mécanismes) a le mérite d'évoquer à la fois le but (comprendre) et les moyens (s'adapter). Elle reste néanmoins trop vague et nous ne saurions nous en contenter.

Préférons-lui celle-ci sous forme de question/réponse. A la question "quelle différence y-a-t-il entre le fonctionnement du cerveau de l'homme et celui des primates", David Krech, auteur d'ouvrages de référence sur le sujet, énonçait la réponse suivante : "Au contaire de l'animal dont les moyens de communication sont uniquement émotionnels, l'homme dispose d'un langage qui a recourt à l'abstraction et la symbolisation". Cette définition de l'intelligence humaine est très intéressante car elle met en exergue les moyens (abstraction et symbolisation) et passe plutôt sous silence le but, la résolution de problèmes. Nous reviendrons à plusieurs reprises sur cette conception de l'intelligence humaine car elle constitue la clef des progrès récents et spectaculaires en matière d'intelligence artificielle.

Sans déflorer le sujet que nous aurons tout loisir de développer plus loin, disons tout de même que la résolution de problèmes en informatique s'appuie sur deux disciplines fondamentales : l'algorithmique et l'heuristique. Alors que la première privilégie le problème à résoudre, souvent au détriment des moyens, la seconde tente de dégager des règles de recherche, indépendantes du contexte et de son objet, c'est-à-dire du problème. Les opérations qui mènent au résultat deviennent alors symboliques (SI le cœur bat trop vite ALORS s'inquiéter) : nous sommes en plein dans le domaine de l'abstraction et de la symbolisation chères à D. Krech.

L'ordinateur "s'inquiète" et agit en fonction... L'intelligence humaine, ses mécanismes, sa nature même, posent une quantité de questions sans réponse. L'intelligence artificielle pose elle une quantité de problèmes à résoudre. Car si les définitions de l'intelligence artificielle brillent plus par leur nombre que par leur rigueur, toutes admettent le même objet : faire simuler par des machines les comportements humains dans la résolution des problèmes. La reconnaissance de formes (pgm RECFORM), l'analyse du langage (SEMAN), la création du mouvement (ROBOT), l'apprentissage (TIC-TAC-TOE et PUISSANCE 4), les systèmes experts (BALANCE), et bien d'autres programmes présentés dans cet ouvrage se

rapportent (quoiqu'à des niveaux différents) à cette catégorie de problèmes, tout à la fois compliqués et tellement passionnants à résoudre : l'imitation des comportements de l'homme avec pour but ultime égaler voire dépasser ses performances. "Imiter", vous l'aviez compris, reste le mot clef pour parler d'intelligence artificielle. Il faut imiter l'intelligence humaine, ce qui n'interdit en aucun cas de faire mieux ! A ce stade, la nécessité de formalisation du processus de l'intelligence de l'homme apparaît clairement. Car pour insuffler une étincelle "d'intelligence" à l'ordinateur, lui donner une capacité de raisonner équivalente à celle de l'homme, la méthode la plus simple consistera à lui inculquer les mêmes principes qui guident l'homme face à un problème à résoudre.

Attention ! La méthode la plus simple ne veut pas dire la meilleure : il est possible, voire probable, que d'autres approches de l'intelligence artificielle puissent donner de meilleurs résultats. Nous attendons simplement le petit génie qui les définira... La rigueur étant de règle dès que l'on s'adresse à des machines qui ne comprennent guère autre chose que des nombres, et encore placés dans un ordre précis, il va falloir au maximum extirper de la démarche humaine tout ce qui est systématique. Ce que nous n'aurons pas pu expliquer (la part de mystère de l'homme) et donc enseigner à l'ordinateur, celui-ci le compensera par sa vitesse de calcul : en quelque sorte, la force brute au secours du manque d'intuition. En détaillant la complexité du processus qui conduit l'homme de la collecte d'informations jusqu'à l'action issue de sa réflexion, on comprendra mieux pourquoi aucun ordinateur ne peut (à ce jour) prétendre l'égaler. En décortiquant ce processus pas à pas, en montrant que chacune des étapes de l'intelligence humaine peut être reproduite par l'intelligence artificielle surtout en donnant pour chacune de ces étapes un exemple de programme d'intelligence artificielle, nous montrerons qu'il n'est pas utopique d'imaginer (un jour lointain) la naissance de l'Androïde cher à Isaac Asimov, parfaite imitation de l'homme.

LES DOMAINES DE L'INTELLIGENCE ARTIFICIELLE

Il y a deux façons (au moins) de parler de l'intelligence artificielle. La première consiste à se gargariser de mots ronflants et évocateurs des mystères de l'informatique. On parlera de reconnaissance de formes (RF), reconnaissance de la parole (Rpa), systèmes experts (SE), bases d'images, de faits, de données, de règles, bases hiérarchiques, relationnelles, etc. Bien sûr, ces mots ont un sens, qui le plus souvent cache des concepts très élaborés et difficiles à appréhender.

Mais les énumérer, même en les expliquant, ne saurait suffire à justifier leur présence dans un ouvrage traitant de l'intelligence artificielle. Pas plus que nous ne saurions ici nous contenter d'affirmer que tous les domaines cités entrent dans le cadre de la seule définition acceptable de l'intelligence artificielle : "imitation des comportements humains dans la résolution de problèmes". Certes, et nous le développerons plus loin, ces domaines s'inspirent bien du modèle humain et c'est quand même là leur principale caractéristique. Mais quelle relation les lie entre eux ? Quel rapport entre la reconnaissance de formes, un système expert, un robot, un jeux électronique d'échecs ? Ces mots n'ont-ils en commun qu'un doux relent d'exotisme scientifique ? Certainement pas. Comme nous allons le voir, tous ces domaines interviennent à un niveau ou à un autre dans le processus global d'intelligence. Ce processus nous allons le décrire en établissant (comme annoncé précédemment) un parallèle entre l'intelligence humaine et artificielle. Plus tard, et comme nulle démonstation ne vaut l'exemple, nous présenterons pour chacune des étapes fondamentales du processus, une application inédite sur Amstrad.

UNE VUE D'ENSEMBLE DE L'INTELLIGENCE HUMAINE

Comment fonctionne l'intelligence humaine, d'un point de vue macroscopique ? Tout commence au niveau des sens (vue, ouïe, etc.). Toutes sortes de signaux en provenance du monde extérieur sont transmis au cerveau qui les filtre, les analyse, les compare, les interprète... Cette première étape de l'intelligence est la PERCEPTION. Puis les données ainsi recueillies sont compilées, transformées, manipulées, mixées à d'autres données déjà dans la mémoire. De nouvelles données sont ainsi produites pendant cette phase, déduite des premières au cours d'un processus itératif complexe (hypothèse, erreur et retours en arrière) pour conduire à une conclusion : c'est la phase de REFLEXION. Enfin ces conclusions peuvent consister en la nécessité d'une intervention sur le monde extérieur : c'est la phase d'ACTION.

Perception, Réflexion, Action sont les trois mamelles de l'intelligence. Les deux étapes aux extrémités sont en contact avec l'environnement. La perception consiste à enregistrer l'état de cet environnement, l'action à le modifier. Entre les deux, la réflexion, typiquement introspective, manipule les données. Tous les exemples tirés de la vie courante, même les plus triviaux, comportent ces trois étapes.

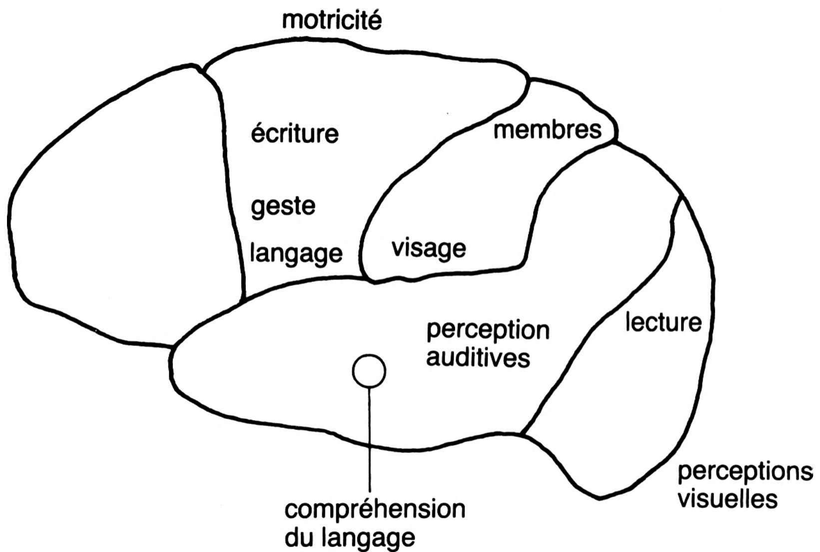
Ainsi, détaillons le processus qui vous conduit à acquérir cet ouvrage. Première étape, PERCEPTION : vous apercevez le livre dans un magasin, vous le feuilletez, vous relevez son prix. Toutes ces données sont adressées à votre cerveau par vos sens : la vue (apercevoir, relever) et l'ouïe (on peut vous dire le prix, l'ambiance sonore du magasin, etc.). L'étape de perception consiste surtout à reconnaître ces données pour ce qu'elles sont. Un rectangle : le livre, les traits sur ce rectangle : des chiffres, ces chiffres : un prix etc. La seconde étape, la REFLEXION, vous amène à rapprocher la nature de l'objet (un livre), ses caractéristiques (intéressant, instructif), de la possibilité de l'acheter (le magasin, son prix raisonnable), de votre désir de le posséder (donnée interne ou déduite des précédentes). Pendant cette phase, un raisonnement vous conduit à la conclusion de l'achat du livre. La dernière étape, l'ACTION, consiste à mettre en œuvre cette conclusion : prendre le livre, aller à la caisse, payer... (A propos, toutes nos félicitations pour cette acquisition !).

En fait, le processus réel est plus compliqué que cela car itéractif : il existe un va-et-vient entre les trois étapes citées. La conclusion et l'action n'interviennent qu'après un grand nombre d'itérations de type Perception/Réflexion/Action. Cependant, on retrouve toujours ces trois étapes, et il sera donc très intéressant de distinguer les mêmes en intelligence artificielle.

LA CARTE DE L'INTELLIGENCE HUMAINE

Nous avons détaillé le processus : avant de l'appliquer à l'intelligence artificielle, examinons comment il est mis en œuvre chez l'homme. Une approche riche en enseignements pour une application future à l'ordinateur consiste à analyser l'intelligence humaine en se plaçant à la frontière entre le macroscopique (le processus) et le micoscopique (neurones et autres cellules nerveuses). Avec cette approche, on peut dresser une véritable carte des fonctions assurées par notre cerveau. Il est remarquable de constater ce découpage physique entre des fonctions déjà difficiles à séparer d'un seul point de vue philosophique.

Les fonctions du cerveau



L'INTELLIGENCE ARTIFICIELLE IMITE L'INTELLIGENCE HUMAINE

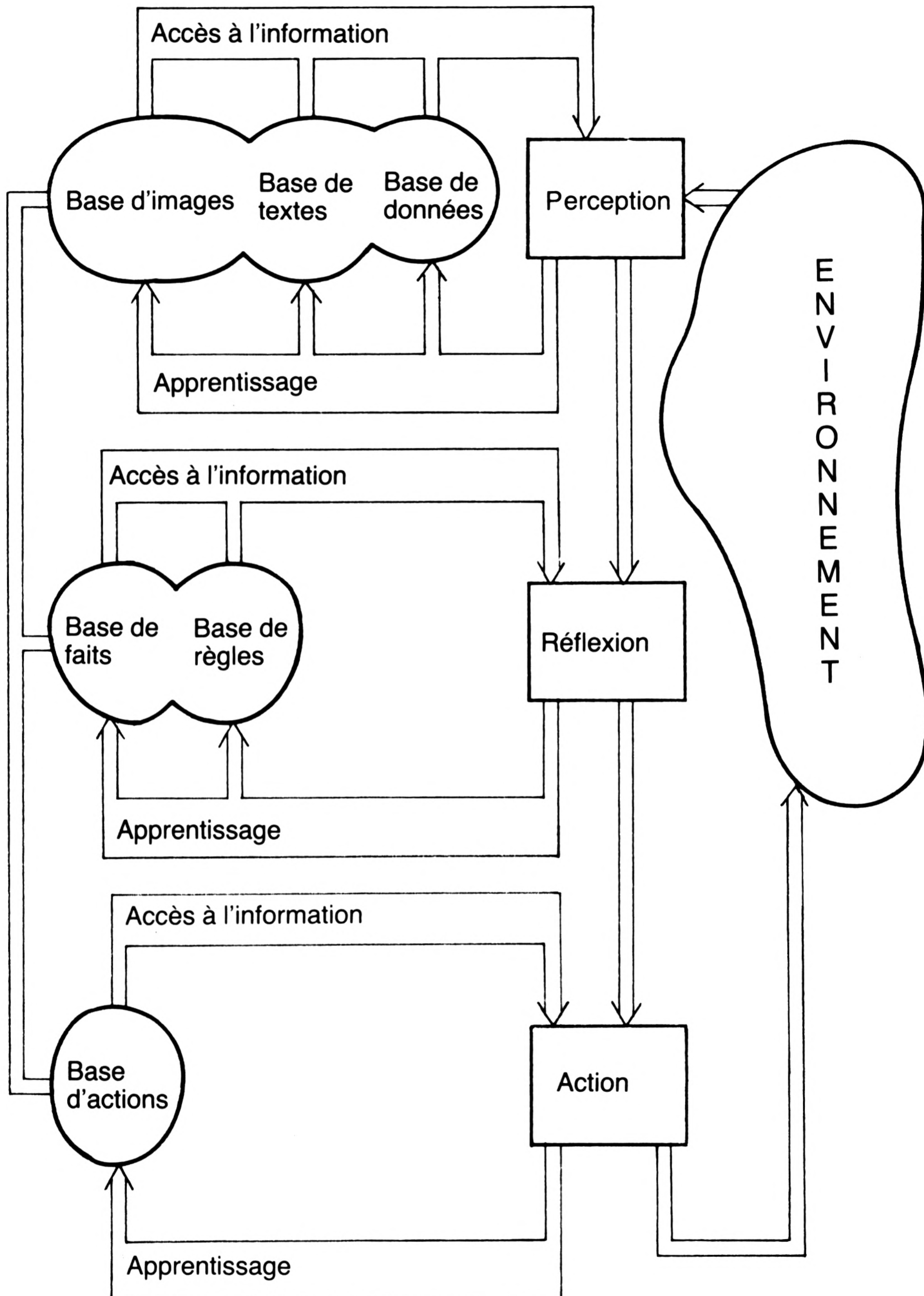
Il est tout à fait possible d'établir un parallèle entre le cerveau humain et l'ordinateur. Les deux peuvent être analysés comme des "systèmes" hiérarchiques. On distingue grossièrement quatre niveaux de hiérarchie (Marr et Poggio, 1977). Le niveau le plus bas concerne l'aspect microscopique : les transistors et les diodes pour l'ordinateur à comparer avec les neurones et les synapses du cerveau. Le niveau suivant s'intéresse aux assemblages réalisés à partir de ces composants élémentaires : additionneur, multiplieur, mémoires, et même microprocesseur pour l'ordinateur. Le troisième niveau est celui de l'algorithmique, c'est-à-dire la description des suites d'opérations nécessaires pour parvenir à un résultat recherché.

L'algorithme de l'addition décrira à l'homme comme à la machine comment mener à bien cette opération. Enfin le niveau le plus élevé vise les théories mathématiques, de l'information, etc. La théorie de l'addition par exemple expliquera la signification de l'opération, son objet, sont but, etc. Elle est indépendante de la nature des chiffres manipulés (romains ou arabes) contrairement à l'étape précédente. Il est ainsi possible de maintenir la comparaison homme/machine du microscopique ou macroscopique.

PERCEPTION, RÉFLEXION, ACTION EN INTELLIGENCE ARTIFICIELLE

On retrouve les mêmes étapes en intelligence artificielle (simplifiées) que nous avons isolées en intelligence humaine.

De la perception à l'action
Les données (gauche), l'intelligence artificielle (centre) et l'environnement (droite)



ENVIRONNEMENT

Pendant la phase de PERCEPTION, l'intelligence artificielle :

– recueillera toutes les informations en provenance du monde extérieur ;

– consultera des bases d'images, de texte, de données pour identifier ces informations ;

– mettra à jour toutes ces bases au cas où l'information ne serait pas présente en mémoire ;

– passera la main à la fonction "Réflexion".

Pendant la phase de RÉFLEXION, toutes les informations enregistrées serviront à nourrir le "cerveau" de l'ordinateur qui :

– consultera des bases de faits et des bases de règles d'interprétation de ces faits ;

– exécutera des algorithmes ou utilisera des méthodes heuristiques pour parvenir à une conclusion ;

– mettra à jour les bases en fonction de cette conclusion ;

– passera la main à la fonction "Action".

Enfin la phase d'ACTION se chargera de mettre en œuvre les conclusions, s'il y a lieu.

DES EXEMPLES DE PROGRAMMES DANS LES TROIS PHASES

Comme on peut le voir sur la figure 4, les applications ne manquent dans aucune des trois étapes citées. Et elles ne sont jamais triviales ! En particulier, il ne fait surtout pas négliger les phases de perception et d'action qui font partie intégrante du processus de l'intelligence. Faire parler un robot, interpréter une photo, effectuer une traduction automatique, constituent autant d'actes complexes nécessitant toutes les ressources de l'intelligence. Ces diverses fonctions font appel aux techniques les plus sophistiquées de l'informatique, comme nous le montreront dans les chapitres suivants, par l'exemple. Nous aborderons successivement et en détail les trois phases en présentant pour chacune d'elles un ou des programmes opérationnels sur Amstrad.

Perception

Reconnaissance de forme

RECFORM peut reconnaître des figures que vous lui dessinerez parmi les figures que vous lui aurez entrées en mémoire.

Reconnaissance de la parole

SYNTAX génère des phrases correctes du point de vue de la syntaxe et de la sémantique, à partir de mots que vous lui aurez précisés.

SEMAN vous comblera d'aise en soutenant un véritable dialogue car il est capable de "comprendre" le sens des mots.

PHRASES répondra à vos questions. Pour cela, il choisira la phrase la plus adéquate parmi toutes celles que vous lui aurez "apprises".

Réflexion

Algorithme et apprentissage.

PUISSANCE 4 et TIC TAC TOE en 3 dimensions sont deux logiciels de jeux utilisant des algorithmes de réflexion. Mais leur particularité, qui vous séduira, réside dans leur capacité d'apprendre au fil des parties... jusqu'à vous battre ? Rien de tel pour progresser qu'un adversaire qui progresse avec vous !

Système expert

BALANCE vous étonnera par son acuité. Ecrit en étroite collaboration avec des médecins spécialistes et généralistes, il vous fournira, après interrogation de vos symptômes, un diagnostic médical et le raisonnement qui l'a guidé.

Action

Robots qui bougent

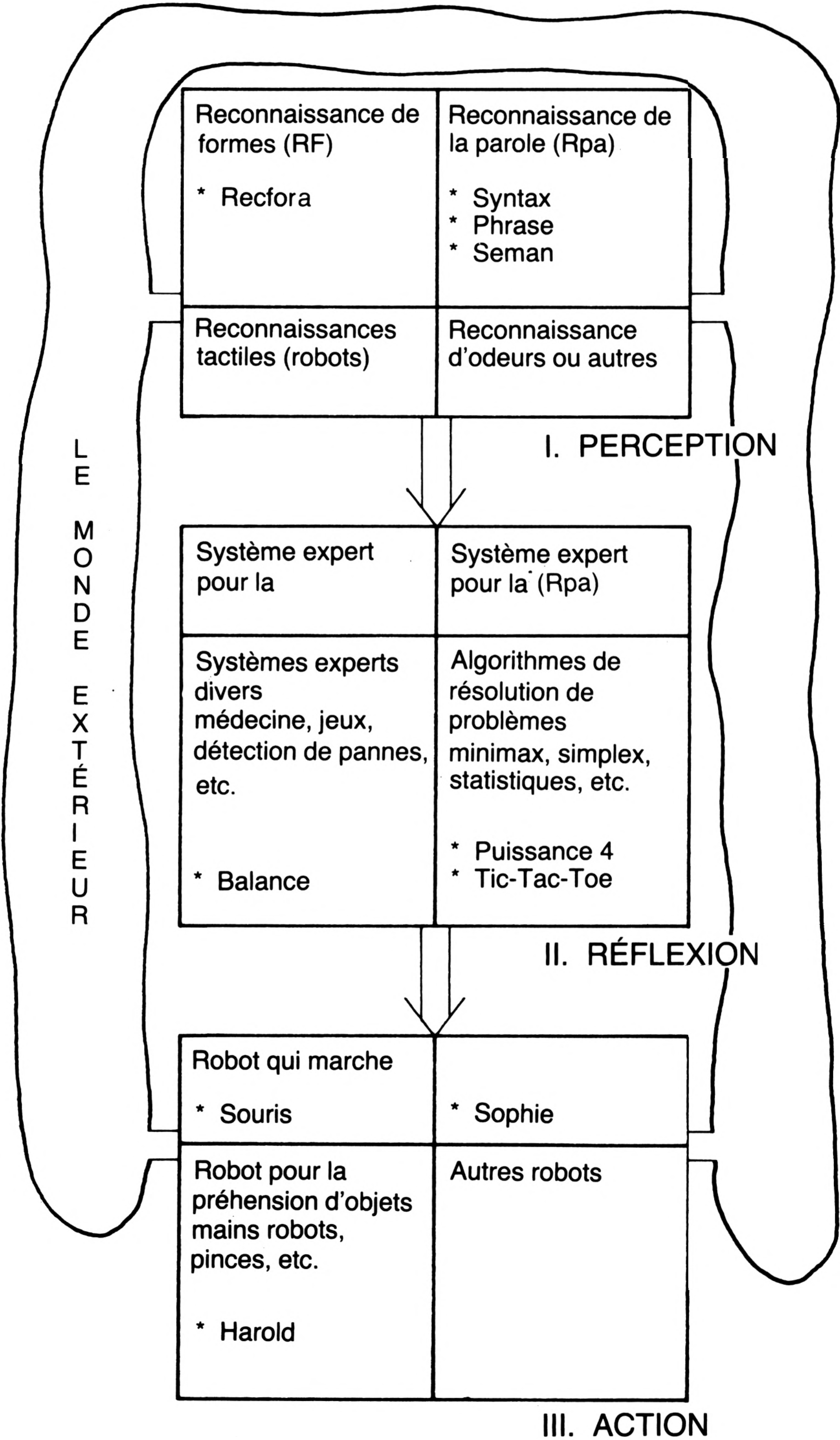
SOURIS trouve la sortie d'un labyrinthe que vous aurez vous-même dessiné.

ROBOT obéit à vos ordres tapés en langage naturel : il évolue dans un décor stylisé et possède un bras terminé par une pince capable de saisir divers objets.

Robots qui parlent

Il vous faudra beaucoup de résistance pour ne pas tomber amoureux de SOPHIE, dont la bouche pulpeuse plus vraie que nature, "prononce" silencieusement un texte que vous tapez à l'écran. Apprenez à lire sur les lèvres de Sophie et rêvez à une femme silencieuse et belle comme... un ordinateur !

Les trois étapes de l'intelligence par l'exemple



Les dix programmes (*) proposés dans cet ouvrage couvrent la plupart des domaines de l'intelligence artificielle.

Attention !

La frontière entre ces programmes est loin d'être aussi marquée qu'il n'y paraît au travers de la classification que nous vous avons proposée. Et cela pour deux raisons. La première tient au fait que le système complet d'intelligence artificielle présenté sur la figure 3 n'est en réalité qu'une vue de l'esprit : il n'existe pas tel quel, mais plutôt morcelé en une multitude de programmes traitant l'un ou l'autre aspect de l'intelligence. Or, pour qu'un programme de reconnaissance de la parole comme SEMAN soit opérationnel, et bien qu'il ait trait à la perception, il nécessite un module de réflexion et même d'action (qui se résumera à l'affichage sur l'écran).

De même, le système expert BALANCE commence par interroger le patient : il comprend donc une part de perception pour saisir les réponses. Il y a 10 programmes présentés dans cet ouvrage, nous pourrions pour chacun dégager les trois étapes du processus d'intelligence artificielle. Cependant, chacun possède un point fort dans une seule (quelquefois deux) de ces étapes : perception, réflexion ou action. En cela, notre découpage se justifie complètement.

De même, comme vous pouvez le constater sur la figure 3, deux fonctions supplémentaires interviennent à tous les niveaux du processus : l'accès aux données et l'apprentissage. Pour ces deux fonctions, nous aurions pu prévoir des chapitres spécifiques : comme elles sont présentes dans presque tous les programmes d'intelligence artificielle, nous avons préféré en parler "à l'occasion". Ainsi, le programme PHRASES traite plus particulièrement l'accès aux données alors que les programmes SOURIS, PUISSANCE 4 et TIC TAC TOE mettent l'emphase sur l'apprentissage.

Avec ces dix programmes, tous les domaines de l'intelligence artificielle seront abordés : il est évident que chaque domaine ne sera que partiellement couvert. Cela tient à la complexité qui les fonde, à leur incroyable richesse, à l'inconnu qui constitue encore la plus grande part de l'intelligence artificielle. Mais cela, c'est l'aventure du 21e siècle !

PREMIÈRE PARTIE : PERCEPTION

CHAPITRE 1
RECONNAISSANCE DE FORMES

QU'EST-CE QUE C'EST ?

Ne vous y fiez pas ! Si le concept de reconnaissance de formes et d'images est un des plus simples à appréhender parmi tous ceux de l'intelligence artificielle, les mécanismes qui entrent en jeu dans sa mise en œuvre atteignent des sommets de complexité rarement égalés. La simplicité apparente du concept relève d'une idée fausse à laquelle il convient sans délai de tordre le cou. Pas plus que les cordes vocales ni la langue n'interviennent dans le sens des mots qu'ils produisent, les yeux ne permettent d'attribuer un sens aux images qu'ils perçoivent. Le cerveau, et lui seul, traite l'ensemble des impulsions électriques transmises par la vue pour les transformer en images, avec tout ce que ce mot renferme d'émotions, de références, d'imagination. De nombreuses expériences montrent toute l'étendue de l'intervention de l'intelligence dans la reconnaissance de formes. Une forme imparfaitement close pose à un observateur (à son insu) le problème de sa clôture.

Celui-ci la complète inconsciemment au point de n'être plus capable (à moins d'un effort dirigé) de discerner le point de rupture de la forme. Un phénomène montre spectaculairement le rôle de l'intelligence en reconnaissance de formes : les analphabètes ne parviennent pas à appréhender correctement les volumes. En effet, le cerveau a besoin d'être entraîné pour être capable de faire sa mise au point légèrement en deçà de l'image et la saisir ainsi en son entier d'un seul coup d'œil.

De toutes les fonctions assurées par notre cerveau, la reconnaissance de formes est une des plus difficiles à analyser et à expliquer en termes concrets. Termes concrets qui pourtant constituent une base de travail indispensable aux informaticiens pour pouvoir reproduire ces mécanismes. On peut admettre, au moins dans le principe, qu'un ingénieur peut dupliquer n'importe quel mécanisme biologique, quand il est parfaitement compris par les neurophysiologistes. On peut admettre en contrepartie que bien rares sont les mécanismes biologiques qui acceptent de livrer leurs secrets, et ceux de la reconnaissance de formes plutôt moins que les autres. Perspectives, couleurs, formes, mais aussi émotion, référentiel culturel, expérience, concourent à optimiser l'interprétation de l'image, formant un réseau complexe et quasiment inextricable... donc très difficile à imiter en intelligence artificielle.

Avant longtemps, les systèmes de reconnaissance de formes ne seront que des ersatz de la merveille que constitue leur modèle humain. Ils pourront en général être facilement pris en défaut et l'indulgence sera de règle en ce qui concerne les rares systèmes de reconnaissance de formes non spécialisés et qui se révèlent les plus ardus à définir.

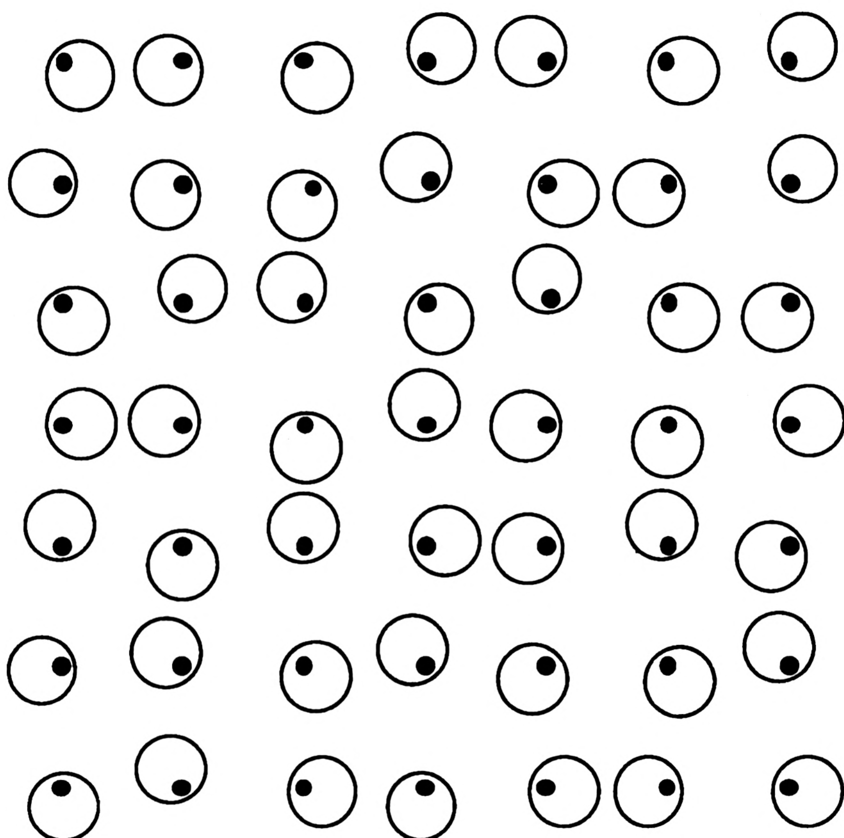
L'ÉTENDUE DU PROBLÈME

Pour ce que l'on comprend des mécanismes de la reconnaissance de formes chez l'homme, il semblerait que les neurones répondent sélectivement à certains stimulis. Parmi eux, citons la taille d'un objet, sa couleur, sa forme, son orientation, etc.

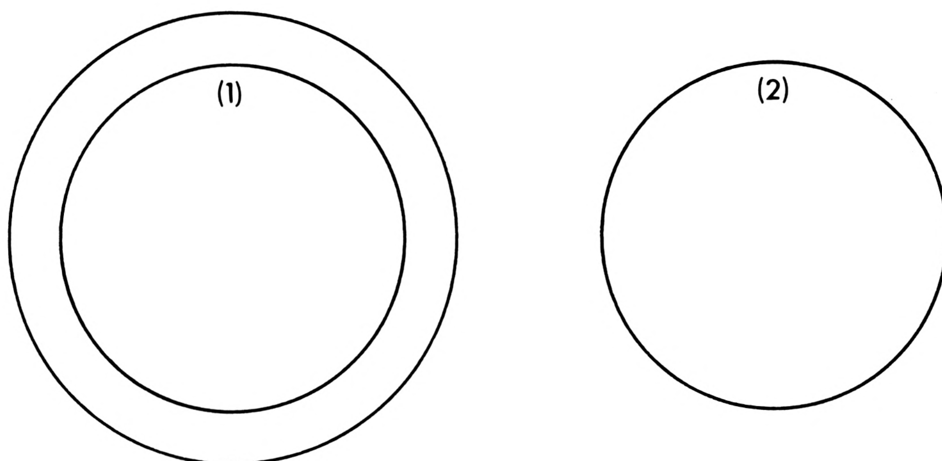
Si l'on se réfère à la figure 1, p. 14 donnant une carte "géographique" des principales fonctions réalisées par le cerveau humain, on peut constater que la reconnaissance de formes et de couleurs y occupe un espace très important. Indiscutablement, dans l'état actuel des recherches en intelligence artificielle et en reconnaissance de formes, il faut bien reconnaître à l'homme une infinie supériorité sur la machine, du point de vue de la simple perception de l'image. L'œil humain est capable de détecter les signaux les plus ténus, puis de les mettre en forme et de les transmettre au cerveau à la vitesse incroyable de plus de 4 millions de bits par seconde ! Quant à l'ensemble du traitement effectué par notre cerveau pour passer d'une succession de signaux électriques à une représentation quasi parfaite de la réalité, il reste en grande partie un mystère pour les neuro-physiologistes, un miracle d'efficacité.

Est-ce à dire que la capacité que possède l'homme de reconnaître les objets qui constituent le monde extérieur est inimitable ? Nous n'affirmerons pas cela, d'autant plus que le but de cet ouvrage est de présenter des programmes d'application de tous les domaines de l'intelligence artificielle y compris la reconnaissance de formes. Il est clair qu'aujourd'hui, aucun système d'intelligence artificielle ne peut prétendre égaler les performances humaines dans ce domaine. Cependant, il existe des situations bien spécifiques où l'ordinateur parvient à surclasser l'homme. Ce dernier peut être par exemple victime "d'illusion d'optique", en réalité victime de certains principes qui fondent ses mécanismes de reconnaissance de formes. Ainsi, si l'on observe la figure 1, on se rend compte que "l'œil" ne parvient pas à discerner la structure formée par les points, pourtant parfaitement régulière, alors qu'il reconnaît la forme des cercles entourant ces points. De même sur la figure 2, l'observateur a bien du mal à admettre que les deux lignes ont la même longueur, ou que les deux cercles sont de taille identique.

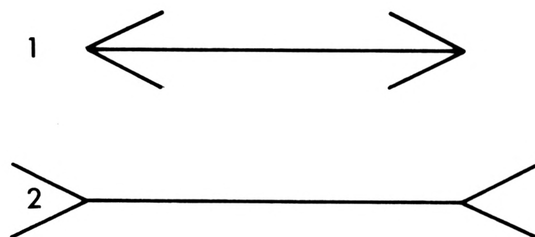
Notre propos n'étant pas d'expliquer en détail les mécanismes de la reconnaissance de formes chez l'homme, nous ne nous étendrons pas sur le pourquoi et le comment de ces quelques faiblesses qui peuvent nous amener à percevoir une image un peu déformée de la réalité. Notons toutefois que ces faiblesses pourront être autant de points forts de l'ordinateur qui dans certains cas pourra se révéler plus efficace que l'homme.



L'œil ne parvient pas à percevoir le réseau de points blancs à cause de leur position excentrique dans les cercles.



Illusion de Delbeuf : les cercles (1) et (2) semblent de tailles différentes.



Illusion de Müller-Lyon :
les segments (1) et (2) semblent de longueurs différentes.

LES DOMAINES D'APPLICATION

Sans produire une liste exhaustive de tous les domaines de recherche et d'application de la reconnaissance de formes, citons ceux qui nous paraissent les plus dignes d'intérêt à court terme car économiquement viables.

Nous voulons parler des domaines suivants :

* militaire : reconnaissance aérienne, visée,... ;
* sécurité : identification de personnes dans l'industrie, les aéroports, les banques, les laboratoires de recherche, etc. ;
* industrie (automobile et autres) : reconnaissance des pièces sur des chaînes, vision des robots d'assemblage,... ;
* médecine : reconnaissance de radiographies, tomographie, etc. ;
* bureautique : tri de documents ;
* pétrole : recherche de nappes pétrolifères par analyse de photos ;
* chimie : identification de produits par analyse de spectrographie.

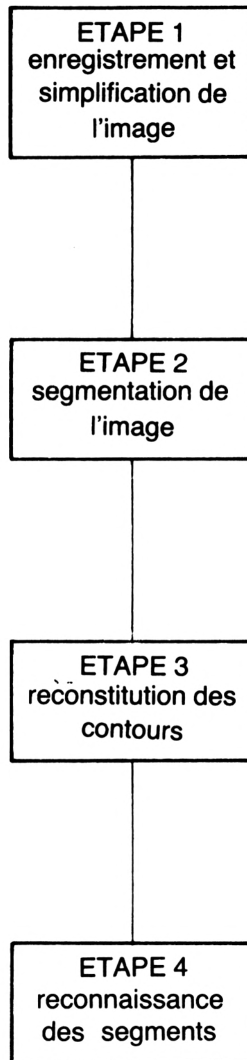
LE PRINCIPE

Les neuro-physiologistes ont appris énormément de choses sur l'aspect microscopique de la vision, le type des cellules qui entrent en jeu, leur capacité de filtrage, leur aptitude à sélectionner certains signaux, etc. mais fort peu pour tout ce qui touche aux mécanismes de la reconnaissance de formes. Or ce sont ces mécanismes qu'il faut comprendre pour pouvoir les imiter avec quelque efficacité en intelligence artificielle.

Il convient d'aborder la reconnaissance de formes sous l'angle du traitement de l'information : c'est par ce biais qu'il sera possible de construire un système de reconnaissance de formes réellement intelligent et donc opérationnel. Dans cette optique, on peut diviser le processus de la reconnaissance de formes en un certain nombre d'étapes fondamentales. Mais avant d'effectuer le découpage qui permettra de classer les problèmes pour mieux les résoudre (diviser pour régner !), tâchons de voir en quoi consiste cette "information" à traiter.

Les étapes

On distingue essentiellement quatre étapes :



Prenons une image des plus simples : un avion dans le ciel. Sur quel type d'informations va pouvoir s'appuyer l'intelligence artificielle pour identifier cet objet ?

la forme,
le fond,
le contexte social,
les circonstances de l'observation,
la référence à d'autres images semblables.

La forme est constituée du fuselage, des ailes, de l'empennage... Le ciel plus quelques nuages occupent le fond de l'image. Notons que ce fond peut éventuellement cacher une partie de la forme. Le contexte social apprend à l'ordinateur l'existence d'avions, généralement sur fond bleu, se déplaçant en ligne droite.

Comme on s'en doute, la part de telles informations dans l'identification de l'objet est prépondérante : un homme de Cromagnon serait bien incapable de reconnaître un avion, et compte tenu de son propre contexte social, le confondrait probablement avec un oiseau géant. Les circonstances de l'observation peuvent apporter une aide précieuse : "en levant les yeux", à l'extérieur, etc. Enfin, et ce point rejoint en partie le contexte social, l'identification ne pourra être effectuée qu'après comparaison avec d'autres images semblables.

LES MOYENS

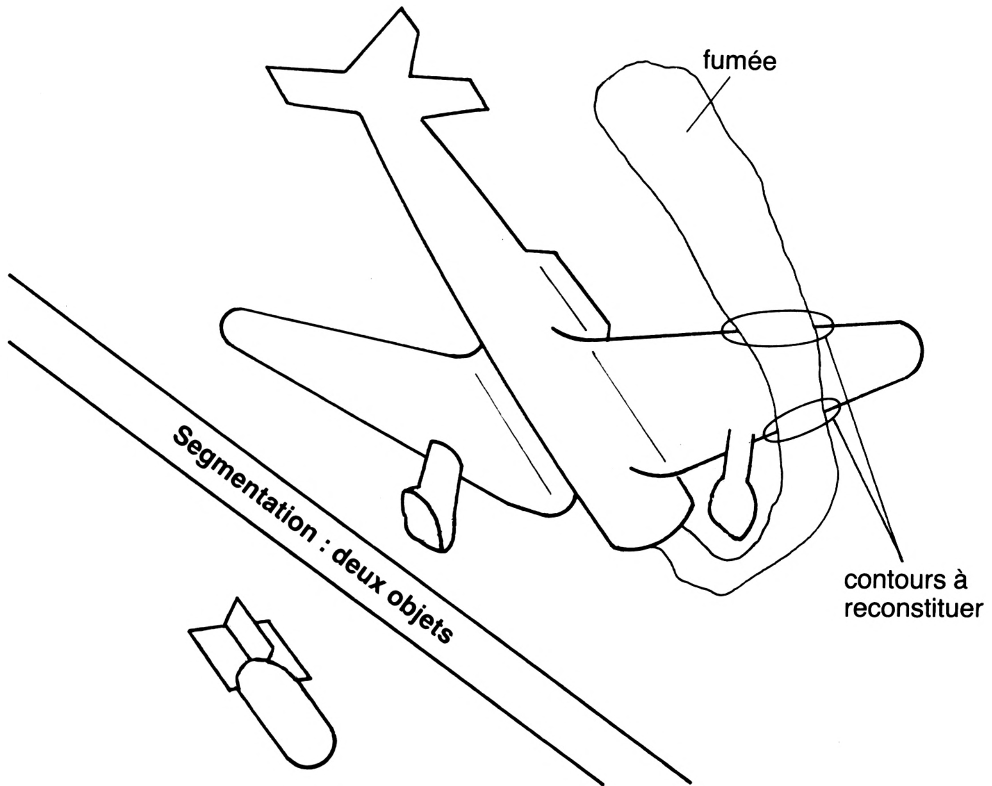
Pour être le plus efficace possible, l'ordinateur devra faire appel aux trois disciplines de base de l'intelligence artificielle :

- l'algorithmique,
- l'heuristique,
- la statistique.

En schématisant, disons que l'algorithmique sert surtout à simplifier l'image, en éliminant les points abérrants, les objets peu intéressants, etc. Des algorithmes serviront à découper une image trop complexe en plusieurs images élémentaires, ou encore à compléter des formes non closes par suite de perturbations extérieures (nuages cachant une partie du fuselage), etc. L'heuristique sera utilisée plutôt pour résoudre des problèmes de type "expertise", par exemple pour vérifier une hypothèse. Le système expert CRYSALIS (1979) détermine la structure de protéines à partir de cartes de densités d'électrons, à trois dimensions. Enfin, la statistique permet de comparer l'image analysée avec des images de référence.

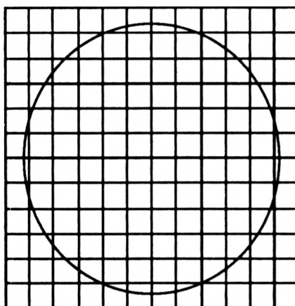
Son utilité apparaît comme une évidence dans les applications médicales de reconnaissance de formes. Un système statistique mis au point à l'Ecole Centrale à Lyon compare des radiographies de colonnes vertébrales déformées avec les mêmes, saines. Grâce à cette comparaison statistique très "fine", et aux conclusions produites par l'ordinateur, un chirurgien peut estimer l'opportunité d'une opération avec un risque d'erreur réduit à sa plus simple expression.

Durant la phase d'enregistrement de l'image, toute une panoplie de techniques intervient dans le but de simplifier au maximum l'image et de ne conserver finalement en mémoire que l'information utile. Ces techniques aussi diverses que variées, peuvent aller du traitement optique de l'image jusqu'à l'application d'algorithmes simples ou très élaborés. Parmi les algorithmes élémentaires, mais qui donnent néanmoins d'excellents résultats, citons le quadrillage de l'image présenté sur la figure 7 ci-après.



Le but est de reconstituer la forme intégrale de l'avion. Il est donc nécessaire d'éliminer la fumée.

L'image



Image

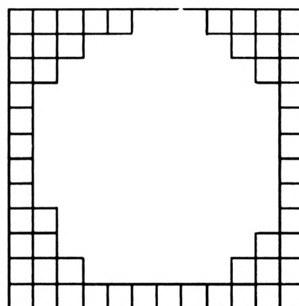


Image après quadrillage

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Image représentée par une matrice 12 × 12

Dans ce cas (cercle), une technique beaucoup plus payante aurait consisté à rechercher l'équation de la forme : $x2 + y2 = a2$. Une opération très spectaculaire dans le cas d'images complexes consiste à rechercher les changements d'intensité de l'image. Aux passages d'une zone d'ombre à une zone éclairée, une ligne est tracée comme sur la figure 8 ci-dessous. Cette technique, mise au point par Marr et Heldreth en 1979, s'avère très rentable pour mémoriser une image dans un espace mémoire minimal.

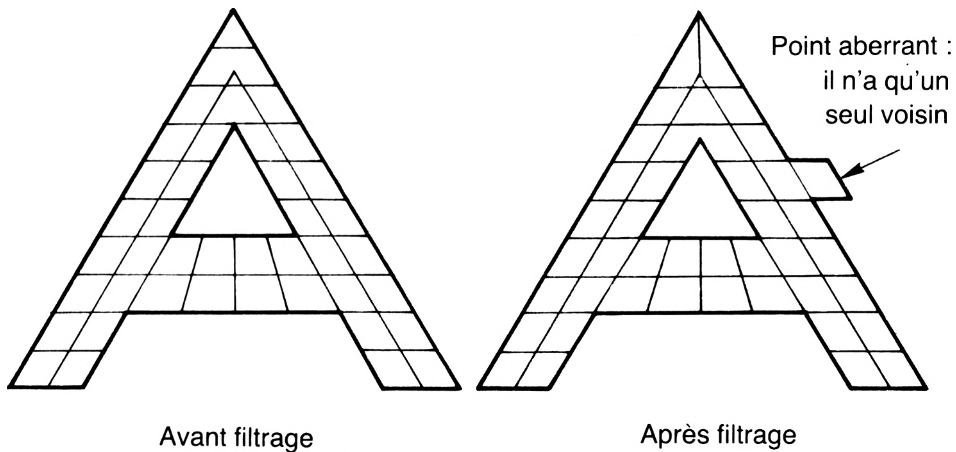


Reconnaissance de formes

Première étape de la reconnaissance de formes : on enregistre les changements d'intensité de l'image (Marr et Hildreth, 1979).

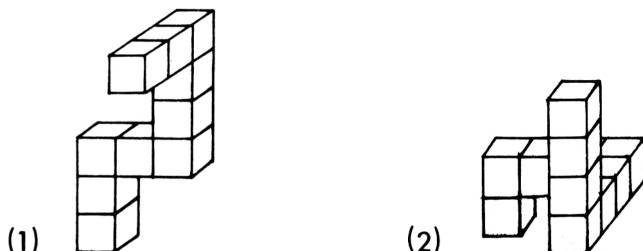
La segmentation consiste à dénombrer puis isoler les différents objets présents dans l'image. Implicitement interviennent là aussi diverses techniques de simplification de l'image : seuls les objets significatifs seront conservés. Cette phase de la reconnaissance de formes voit son efficacité grandement accrue si le programme sait ce qu'il cherche : par exemple une horloge dans une place d'église, etc. Ainsi, le programme de Freuder identifie un marteau dans une scène complexe.

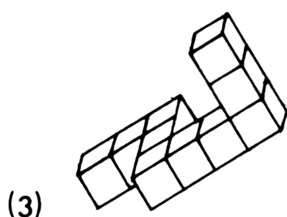
Durant la phase de reconstitution des contours, l'ordinateur cherche à obtenir des images le plus net possible. Par exemple, un algorithme de "filtrage" tentera d'éliminer les points aberrants, c'est-à-dire n'ayant pas au moins deux points voisins : cette technique corrige certaines erreurs intervenues au moment de la saisie de l'image, due à des parasites. D'autre part, certaines lignes accidentellement coupées sont reconstituées.

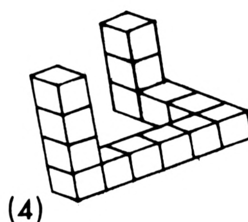


Seuls les points possédant plus d'un voisin sont conservés.

La reconnaissance proprement dite de l'image n'intervient qu'après toutes ces étapes préalables. On dispose alors d'une image "saine" que l'on peut comparer avec des images de référence sans être gêné par d'autres images parasites ou simplement mal saisies. Toutes les erreurs de ce type sont censées avoir disparu à ce stade. Cette reconnaissance s'effectue soit par des méthodes statistiques (dès qu'il s'agit de juger de la ressemblance avec une référence, les statistiques se révèlent très puissantes), soit par des algorithmes. Ces derniers doivent être capables de "triturer" l'image dans tous les sens, de la déformer, lui faire effectuer toutes les rotations possibles.

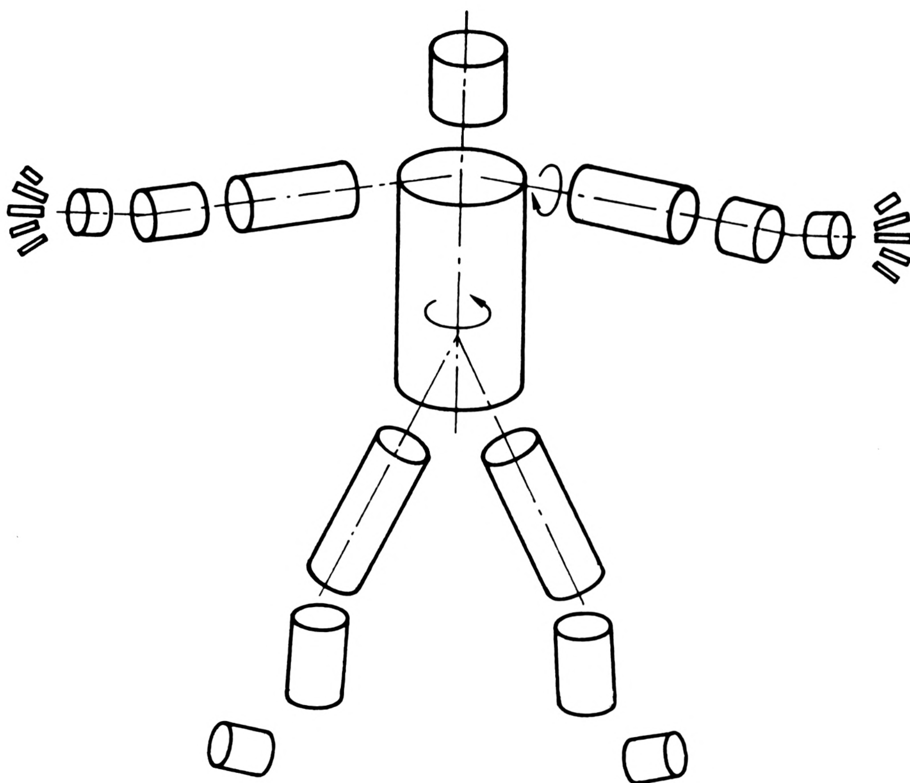






Seules (1) et (3) sont identiques, par rotation dans l'espace.
Aucune rotation n'amène (2) et (4) en convergence.

Ainsi, quelle que soit la position de l'image dans l'espace, sa taille, ses variantes de forme, le programme doit trouver une ressemblance entre elle et l'image de référence enregistrée préalablement. Pour reprendre l'exemple de l'avion, l'intelligence artificielle doit être capable de reconnaître un avion dans n'importe quelle position (au décollage, de dessus, de côté, etc.) en n'ayant qu'une seule image en mémoire. Pour réaliser cela, on fait intervenir la notion de modèle : une description modélisée d'un objet donné, avec ses axes de rotation, les variations possibles de formes, etc. La figure suivante montre un modèle simplifié d'homme.



Modèle 3D de l'homme, avec les axes de rotation. Il permet de reconnaître une image d'homme déduite de la forme initiale par des rotations autour de ces axes.

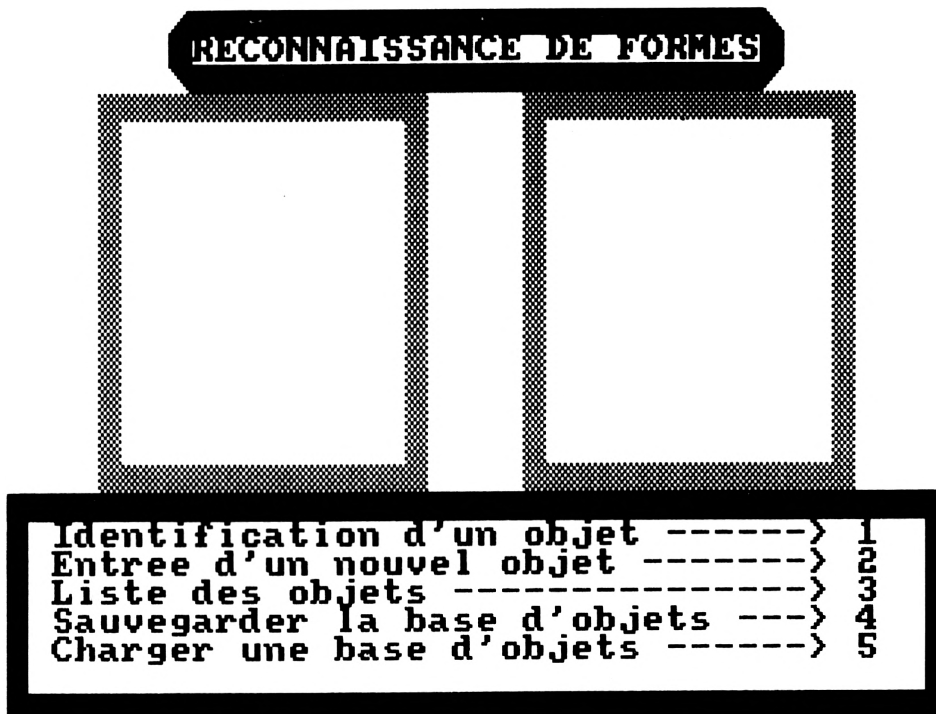
PRÉSENTATION DU PROGRAMME RECFORM

COMMENT L'UTILISER ?

Ce logiciel permet la mémorisation d'une base d'objets (auparavant tracés par l'utilisateur) puis la recognition, parmi ceux-ci, d'un objet tracé par l'utilisateur.

L'écran se présente sous la forme de trois fenêtres. La fenêtre inférieure est dédiée au dialogue. La fenêtre supérieure gauche est celle sur laquelle l'utilisateur dessine une figure, à l'aide de l'utilitaire graphique intégré, tandis que la fenêtre supérieure droite peut afficher un objet contenu en mémoire.

Les cinq options du menu principal permettent :



RECFORM 1 : le menu général vous offre 5 options

– la création d'un nouvel objet : l'utilisateur trace une figure, le logiciel l'analyse et la mémorise ;

– la recognition d'un objet : l'utilisateur trace une figure, puis le logiciel l'analyse et cherche en mémoire l'objet le plus ressemblant, puis l'affiche dans la fenêtre de droite. D'autre part le pourcentage de ressemblance est affiché ;

– liste des objets : les objets en mémoire défilent un à un dans la fenêtre de droite ;

– sauvegarde de la base d'objets ;

– lecture d'une base d'objets.

La création d'une figure s'effectue à l'aide d'un "crayon" pilotable à partir des touches du pavé numérique (12346789). Ce crayon peut être levé, baissé ou agir comme une gomme (il apparaît à l'écran, respectivement sous forme des lettre L, B ou G) ; la commutation entre ces trois modes s'effectue par la touche espace. Lorsqu'une figure est terminée, il suffit de presser la touche ENTER.

COMMENT ÇA MARCHE ?

Le cœur de ce logiciel est la partie analyse des caractéristiques de l'objet nouvellement tracé par l'utilisateur, c'est celle-ci qui permettra ultérieurement la recognition de cet objet.

Cette grande partie est composée de trois modules, dont les fonctions sont respectivement le calcul :

- du nombre de trous,
- du nombre de figures,
- du centre de gravité ainsi que du nombre de pointes et d'angles.

Ces critères de distinction ont été fixés arbitrairement, après l'expérimentation de plusieurs solutions. Nous allons maintenant développer plus en détail le fonctionnement de chacun de ces modules.

Détermination du nombre de trous

Par convention, un trou est une surface composée de points éteints et entièrement entourée de points allumés. Le programme commence par chercher une case vide puis, à la façon du programme d'exploration de labyrinthe, il explore les cases alentour jusqu'à ce qu'il rencontre soit un point allumé, soit un point déjà exploré, auquel cas il explore une autre case (ou revient en arrière en cas d'impasse), soit un bord, auquel cas la surface explorée est ouverte et ne fait donc pas partie d'un trou. Si le retour en arrière ramène au premier point exploré, sans qu'un bord du cadre ait été rencontré alors, la surface explorée est un trou et le nombre de trous (NTROUS) est incrémenté.

Le programme reprend alors le cycle initial : recherche d'un point non exploré et éteint... et ainsi de suite jusqu'à ce que tout le dessin soit exploré.

Signalons que l'exploration d'une surface s'effectue par déplacement dans quatre directions (donc aucun déplacement en diagonale).

Détermination du nombre de figures

Le nombre de figures est le nombre d'objets indépendants, c'est-à-dire le nombre de groupes de points contigus, ces groupes n'ayant aucun point en commun. Ainsi la lettre i est composée de deux figures (le corps et le point).

Ce module fonctionne comme le précédent, à ceci prêt qu'il effectue la recherche de surfaces de points allumés au lieu des points éteints. Autres différences, la rencontre d'un bord est assimilée à celle d'un point éteint et les déplacements, durant la recherche peuvent s'effectuer en diagonale.

En fait, ces deux modules sont assez identiques et auraient pu être regroupés en un seul mais nous avons préféré, dans un souci de clarté, réaliser deux parties distinctes.

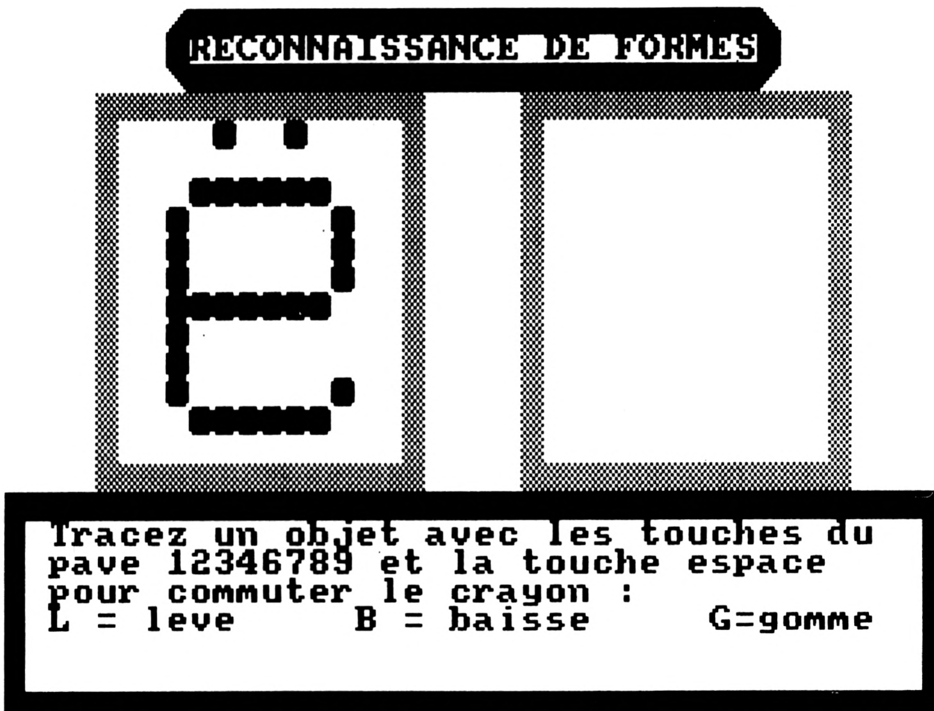
Détermination du centre de gravité, du nombre de pointes et d'angles

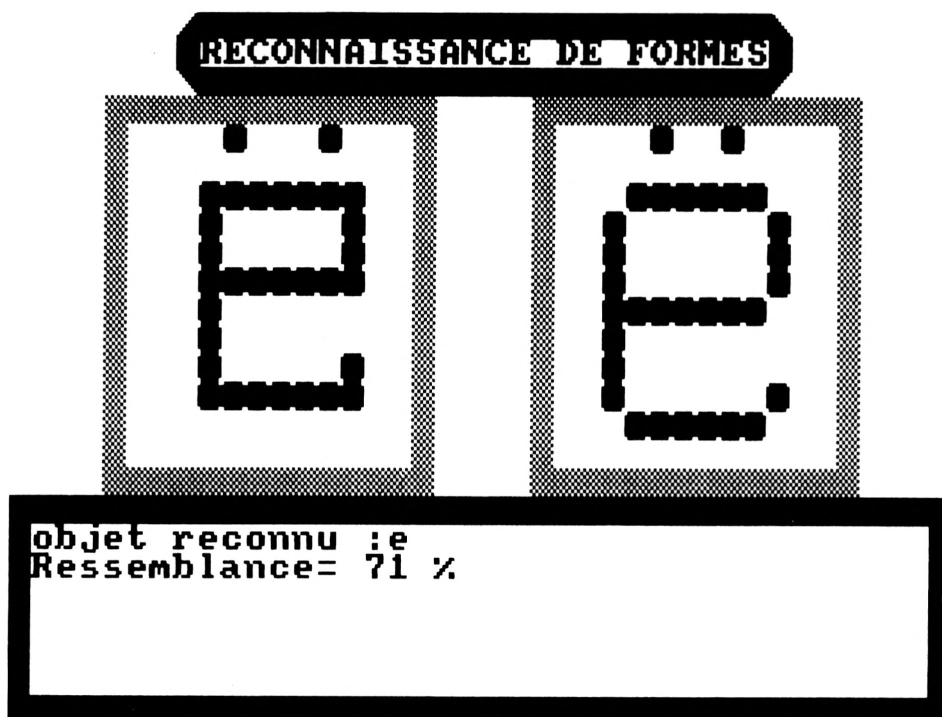
Les coordonnées du centre de gravité sont calculées sous forme de fractions de la largeur et de la hauteur de l'objet, elles ne dépendent donc pas de la position de cet objet dans le cadre. Ce centre est rapporté au système dont l'axe des X est horizontal et tangent au bord inférieur de la figure, et dont l'axe des Y est vertical et tangent au bord gauche de cette même figure.

Tout point qui est adjacent à un seul point est, par convention, une "pointe", ainsi un A majuscule possédera deux pointes (les deux jambes).

Enfin un angle est caractérisé par trois points adjacents non alignés.

Lorsque l'utilisateur trace un objet dans le but de le faire reconnaître par le logiciel, celui-ci le compare à chacun des objets en mémoire (module reconnaissance de l'objet tracé). Pour chaque objet, critère par critère, l'écart entre celui-ci et l'objet tracé est calculé, puis la somme des écarts (affectés de coefficients d'importance) est effectuée. L'objet ayant obtenu la plus petite somme est considéré comme l'objet le plus ressemblant, il est affiché, ainsi que le pourcentage de ressemblance.





RECFORM 2 et 3 : l'option "Entrée d'un nouvel objet" a été choisie. Cet objet contient 1 pointe, 1 trou, 3 figures indépendantes et 4 angles. Malgré la déformation de l'objet, celui-ci est reconnu.

Précisons la façon dont cet écart entre deux objets est calculé : soit l'objet 1 possédant NTROUS1 nombre de trous, NPOINTES1 nombre de pointes, NFIGURES1 nombre de figures, NANGLES1 nombre d'angles, et un centre de gravité de coordonnées GX1 et GY1. De même l'objet 2 possède NTROUS2 nombres de trous, NPOINTES2 nombre de pointes, NFIGURES2 nombre de figures, NANGLES2 nombre d'angles, et un centre de gravité de coordonnées GX2 et GY2. L'écart entre l'objet 1 et l'objet 2 est alors calculé pour chaque critère :

ECART (NTROUS)=ABS(NTROUS1-NTROUS2)

ECART (NPOINTES)=ABS(NPOINTES1-NPOINTES2)

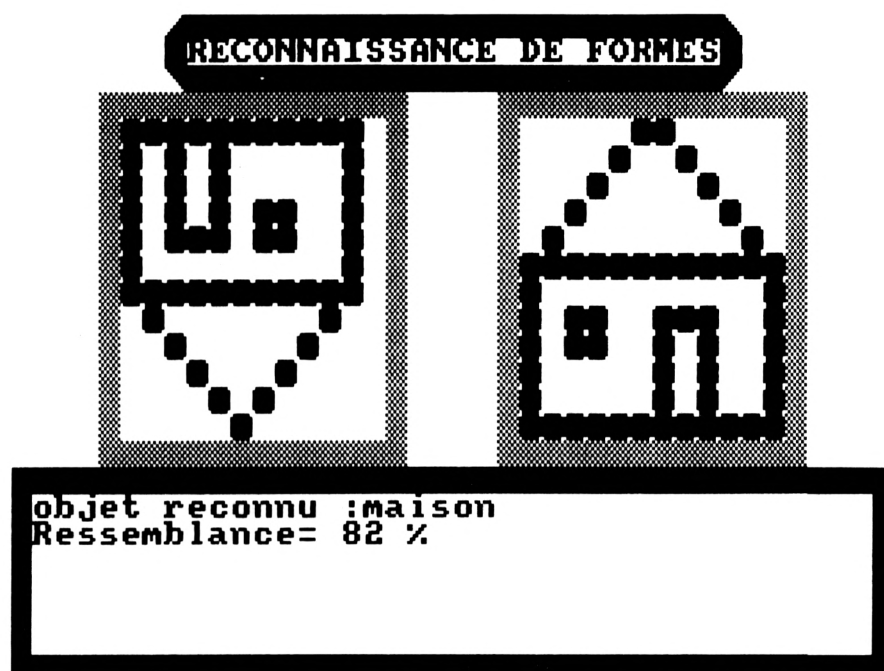
ECART (NFIGURES)=ABS(NFIGURES1-NFIGURES2)

ECART (NANGLES)=ABS(NANGLES1-NANGLES2)

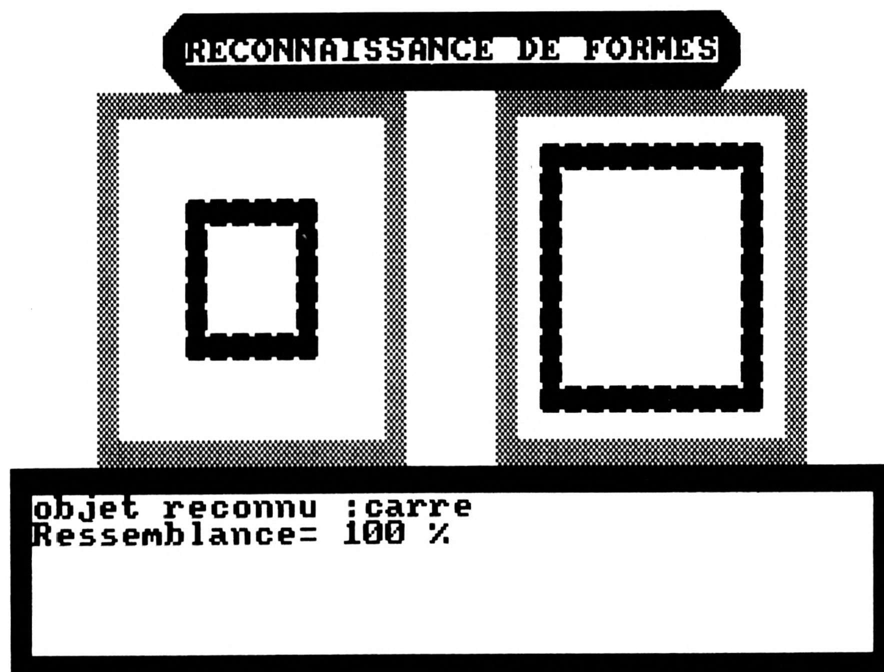
ECART (G)=ABS(GX1-GX2)+ABS(GY1-GY2)

Ensuite l'écart total entre les deux objets est obtenu en effectuant la somme des écarts, ceux-ci étant affectés de coefficients relativisant leur importance, arbitraires, ils ont été déterminés par l'expérience, d'autant que les différents écarts ne sont pas des grandeurs de même nature, donc :

ECARTtotal = 9*ECART(NTROUS) + 9*ECART(NPOINTES) + 9*ECART(NFIGURES) + 2*ECART(NANGLES) + 30*ECART(G)



RECFORM 4 : l'identification de l'objet de la fenêtre de gauche a bien lieu (fenêtre de droite), et cela malgré le fait que la maison ne soit pas exactement identique au modèle, et qu'elle ait été tracée à l'envers.



RECFORM 5 : identification à 100 % de ce carré, malgré la modification d'échelle.

PRINCIPALES VARIABLES

ntrous(i)	nombre de trous de l'objet i
ntrous	nombre de trous de l'objet analysé
npointes(i)	nombre de pointes de l'objet i
npointes	nombre de pointes de l'objet analysé
nfigures(i)	nombre de figures indépendantes de l'objet i
nfigures	nombre de figures indépendantes de l'objet analysé
nangles(i)	nombre d'angles de l'objet i
nangles	nombre d'angles de l'objet analysé
gx(i) et gy(i)	coordonnées du centre de gravité de l'objet i
gx et gy	coordonnées du centre de gravité de l'objet analysé
nom$(i)	nom de l'objet i
point(i)	état de chaque point i de l'objet analysé
pointm(i)	sauvegarde de ces états (l'analyse étant susceptible de les modifier)
chemin(i)	mémorisation du chemin (points i) explorés durant la recherche du nombre de trous ou de figures indépendantes.
pointc(i, 143)	mémorisation des 144 points de l'objet i

```
10 REM********************
20 REM  RECFORM (T. L.-A.)
30 REM********************
40 MODE 1
50 REM===========================
60 REM initialisation des variables
70 REM===========================
80 DEFINT a-z
90 DEFREAL e,g
100 SYMBOL 240,126,255,255,255,255,255,255,126
110 DIM dep(4),point(196),pointm(196),paramf(5),chemin(144)
120 DIM caract$(1),dep8(7)
130 DIM ntrous(40),npointes(40),nfigures(40),nangles(40),nom$(40)
140 DIM gx(40),gy(40),pointc(40,143)
150 dep8(0)=-14:dep8(1)=-13:dep8(2)=1:dep8(3)=15
160 dep8(4)=14:dep8(5)=13:dep8(6)=-1:dep8(7)=-15
170 haut=0:bas=2
180 gauche=3:droite=1
190 arret=4
200 dep(haut)=-14:dep(bas)=14
210 dep(gauche)=-1:dep(droite)=1
220 eteint=0:allume=1
230 inexplore=0:explore=3:bord=4
240 caract$(allume)=CHR$(240):caract$(eteint)=" "
250 leve=1:baisse=0:gomme=2
260 crayon$(leve)="L"
```

```
270 crayon$(baisse)="B"
280 crayon$(gomme)="G"
290 ox=5:oy=4:ox1=23:oy1=4
300 fermee=0:ouverte=1
310 FOR i=0 TO 13
320   FOR i0=0 TO 182 STEP 182
330     point(i+i0)=bord
340   NEXT i0
350 NEXT i
360 FOR i=14 TO 168 STEP 14
370   FOR i0=0 TO 13 STEP 13
380     point(i+i0)=bord
390   NEXT i0
400 NEXT i
410 REM=========================
420 REM initialisation graphique
430 REM=========================
440 WINDOW#0,2,39,19,24
450 WINDOW#2,24,35,5,16
460 WINDOW#3,6,17,5,16
470 LOCATE#1,9,1:PRINT#1,STRING$(24,143);
480 LOCATE#1,9,3:PRINT#1, STRING$(24,143);
490 LOCATE#1,23,4:PRINT#1,STRING$(14,207);
500 LOCATE#1,23,17:PRINT#1,STRING$(14,207);
510 LOCATE#1,5,4:PRINT#1,STRING$(14,207);
520 LOCATE#1,5,17:PRINT#1,STRING$(14,207);
530 FOR x0=5 TO 23 STEP 18
540   FOR x=0 TO 13 STEP 13
550     FOR y=4 TO 16
560       LOCATE#1,x+x0,y:PRINT#1,CHR$(207);
570     NEXT y
580   NEXT x
590 NEXT x0
600 LOCATE#1,8,2:PRINT#1,CHR$(143);
610 LOCATE#1,33,2:PRINT#1,CHR$(143);
620 LOCATE#1,8,1:PRINT#1,CHR$(214);
630 LOCATE$1,33,1:PRINT#1,CHR$(215);
640 LOCATE#1,8,3:PRINT#1,CHR$(213);
650 LOCATE#1,33,3:PRINT#1,CHR$(212);
660 LOCATE#1,9,2:PRINT#1,"RECONNAISSANCE DE FORMES";
670 LOCATE#1,1,18
680 FOR i=1 TO 8:PRINT#1,STRING$(40,143);:NEXT i
690 CLS
700 REM***************************
710 REM  menu et boucle principale
720 REM***************************
730 PRINT " Identification d'un objet ------> 1"
740 PRINT " Entree d'un nouvel objet -------> 2"
```

```
750 PRINT " Liste des objets ---------------> 3"
760 PRINT " Sauvegarder la base d'objets ---> 4"
770 PRINT " Charger une base d'objets ------> 5"
780 choix$=INKEY$
790 IF choix$="1" THEN 860
800 IF choix$="2" THEN 930
810 IF choix$="3" THEN 1000
820 IF choix$="4" THEN 2910
830 IF choix$="5" THEN 3120
840 GOTO 780
850 REM--- reconnaissance d'un objet ---
860 CLS
870 IF nobjets=0 THEN PRINT "Impossible : pas d'objets en memoire":GOTO 730
880 CLS:GOSUB 1100:REM dessin de l'objet
890 GOSUB 1500:REM analyse des caracteristiques de l'objet
900 GOSUB 2260:REM recherche de l'objet le plus ressemblant
910 GOSUB 2790:REM re-initialisation graphique
920 GOTO 730
930 REM--- entree d'un nouvel objet ---
940 CLS:GOSUB 1100:REM dessin de l'objet
950 GOSUB 1500:REM analyse des caracteristiques de l'objet
960 GOSUB 2450:REM enregistrement du nouvel objet
970 GOSUB 2790:REM re-initialisation graphique
980 GOTO 730
990 REM--- liste des objets ---
1000 IF nobjets=0 THEN PRINT"Impossible:pas d'objets en memoire":GOTO 730
1010 FOR objet=0 TO nobjets-1
1020   GOSUB 2710:PRINT "Nom de cet objet : ";nom$(objet)
1030   INPUT "Pour continuer, appuyez sur <ENTER>",rep$:CLS#2
1040 NEXT objet
1050 GOTO 730
1060 REM****************************************
1070 REM==================
1080 REM trace de la figure
1090 REM==================
1100 PRINT" Tracez un objet avec les touches du"
1110 PRINT" pave 12346789 et la touche espace"
1120 PRINT" pour commuter le crayon :"
1130 PRINT" L = leve     B = baisse     G=gomme"
1140 x=1:y=1:xm=1:ym=1:ppoint=15:pmpoint=15:etat=leve:dir$=""
1150 WHILE dir$<>CHR$(13)
1160   REM---test clavier---
1170   dir$="":dep=0:dir$=INKEY$
1180   IF dir$=" " THEN etat=etat+1:IF etat=3 THEN etat=0
1190   crayon=etat
1200   IF dir$="8" THEN dep=dep8(0)
1210   IF dir$="9" THEN dep=dep8(1)
1220   IF dir$="6" THEN dep=dep8(2)
```

```
1230   IF dir$="3" THEN dep=dep8(3)
1240   IF dir$="2" THEN dep=dep8(4)
1250   IF dir$="1" THEN dep=dep8(5)
1260   IF dir$="4" THEN dep=dep8(6)
1270   IF dir$="7" THEN dep=dep8(7)
1280   ppoint=ppoint+dep
1290   REM------------------
1300   IF point(ppoint)=bord THEN ppoint=ppoint-dep
1310   IF crayon=baisse THEN point(ppoint)=allume
1320   IF crayon=gomme THEN point(ppoint)=eteint
1330   IF cligno=0 THEN cligno=1 ELSE cligno=0
1340   y=INT((ppoint-1)/14):x=ppoint-14*y
1350   LOCATE#1,ox+x,oy+y
1360   IF cligno=0 THEN PRINT#1,crayon$(crayon);:GOTO 1380
1370   PRINT#1,caract$(point(ppoint));
1380   LOCATE#1,ox+xm,oy+ym
1390   FOR temps=1 TO 100:NEXT temps
1400   PRINT#1,caract$(point(pmpoint));
1410   pmpoint=ppoint:xm=x:ym=y
1420 WEND
1430 CLS
1440 LOCATE#1,ox+x,oy+y:xm=x:ym=y
1450 PRINT#1,caract$(point(ppoint));
1460 RETURN
1470 REM=======================================
1480 REM analyse des caracteristiques de l'objet
1490 REM=======================================
1500 REM--------------------------------
1510 REM determination du nombre de trous
1520 REM--------------------------------
1530 ntrous=0:pt=0
1540 GOSUB 2640:REM transfert de la figure dans un tableau de travail
1550 ppoint=15
1560 WHILE pointm(ppoint)<>inexplore AND ppoint<>181
1570   ppoint=ppoint+1
1580 WEND
1590 IF ppoint=181 THEN 1730
1600 surface=trou
1610 chemin(pt)=ppoint:pt=pt+1:pointm(ppoint)=explore
1620 dep=arret
1630 FOR dir=0 TO 3
1640   point=pointm(ppoint+dep(dir))
1650   IF point=bord THEN surface=ouverte
1660   IF point=inexplore THEN dep=dep(dir)
1670 NEXT dir
1680 IF dep<>arret THEN ppoint=ppoint+dep:GOTO 1610
1690 IF pt<>0 THEN 1720
1700 IF surface=trou THEN ntrous=ntrous+1
```

```
1710 GOTO 1550
1720 pt=pt-1:ppoint=chemin(pt):GOTO 1620
1730 REM----------------------------------
1740 REM determination du nombre de figures
1750 REM----------------------------------
1760 nfigures=0:pt=0
1770 GOSUB 2640:REM transfert de la figure dans un tableau de travail
1780 ppoint=15
1790 WHILE pointm(ppoint)<>allume AND ppoint<>181
1800   ppoint=ppoint+1
1810 WEND
1820 IF ppoint=181 THEN 1970
1830 chemin(pt)=ppoint:pt=pt+1:pointm(ppoint)=explore
1840 dep=arret
1850 FOR dir=0 TO 7
1860   point=pointm(ppoint+dep8(dir))
1870   IF point=allume THEN dep=dep8(dir)
1880 NEXT dir
1890 IF dep<>arret THEN ppoint=ppoint+dep:GOTO 1830
1900 IF pt<>0 THEN 1930
1910 nfigures=nfigures+1
1920 GOTO 1780
1930 pt=pt-1:ppoint=chemin(pt):GOTO 1840
1940 REM--------------------------------------------------------------
1950 REM determination du centre et du nombre de pointes et d'angles
1960 REM--------------------------------------------------------------
1970 npointes=0:nangles=0
1980 xmin=100:ymin=100:xmax=0:ymax=0:npoints=0:gx=0:gy=0
1990 GOSUB 2640:REM transfert de la figure dans un tableau de travail
2000 FOR x=1 TO 12
2010   FOR y=1 TO 12
2020     ppoint=14*y+x
2030     IF pointm(ppoint)<>allume THEN 2170
2040     IF x<xmin THEN xmin=x
2050     IF x>xmax THEN xmax=x
2060     IF y<ymin THEN ymin=y
2070     IF y>ymax THEN ymax=y
2080     gx=gx+x:gy=gy+y
2090     npoints=npoints+1
2100     nvoisins=0:mdir(0)=0:mdir(1)=0
2110     FOR dir=0 TO 7
2120       point=pointm(ppoint+dep8(dir))
2130       IF point=allume THEN mdir(nvoisins)=dir:nvoisins=nvoisins+1
2140     NEXT dir
2150     IF nvoisins=1 THEN npointes=npointes+1
2160     IF nvoisins=2 AND mdir(1)-mdir(0)<>4 THEN nangles=nangles+1
2170   NEXT y
2180 NEXT x
```

```
2190 IF npoints=0 THEN gx=0:gy=0:RETURN
2200 IF xmax=xmin THEN gx=0 ELSE gx=(gx/npoints-xmin)/(xmax-xmin)
2210 IF ymax=ymin THEN gy=0 ELSE gy=(gy/npoints-ymin)/(ymax-ymin)
2220 RETURN
2230 REM================================
2240 REM reconnaissance de l'objet trace
2250 REM================================
2260 ecartmin=1000
2270 FOR objet=0 TO nobjets-1
2280   ecart=9*ABS(ntrous(objet)-ntrous)
2290   ecart=ecart+9*ABS(nfigures(objet)-nfigures)
2300   ecart=ecart+9*ABS(npointes(objet)-npointes)
2310   ecart=ecart+2*ABS(nangles(objet)-nangles)
2320   ecart=ecart+30*(ABS(gx(objet)-gx)+ABS(gy(objet)-gy))
2330   IF ecart<ecartmin THEN objetbis=objet:ecartmin=ecart
2340 NEXT objet
2350 global=9*ntrous+9*nfigures+9*npointes+2*nangles+6*(gx+gy)
2360 ressemblance=100*(1-ecartmin/global)
2370 REM---affichage de l'objet reconnu---
2380 objet=objetbis:GOSUB 2710
2390 PRINT"objet reconnu :";nom$(objetbis)
2400 PRINT"Ressemblance=";ressemblance;"%"
2410 RETURN
2420 REM===============================
2430 REM enregistrement du nouvel objet
2440 REM===============================
2450 ppoint2=0
2460 FOR y=1 TO 12
2470   FOR x=1 TO 12
2480     ppoint=14*y+x
2490     pointc(nobjets,ppoint2)=point(ppoint)
2500     ppoint2=ppoint2+1
2510   NEXT x
2520 NEXT y
2530 ntrous(nobjets)=ntrous
2540 npointes(nobjets)=npointes
2550 nfigures(nobjets)=nfigures
2560 nangles(nobjets)=nangles
2570 gx(nobjets)=gx:gy(nobjets)=gy
2580 INPUT "Nom de ce nouvel objet";nom$(nobjets)
2590 nobjets=nobjets+1
2600 RETURN
2610 REM==============================================
2620 REM transfert de la figure dans un tableau de travail
2630 REM==============================================
2640 FOR ppoint=0 TO 195
2650   pointm(ppoint)=point(ppoint)
2660 NEXT ppoint
```

```
2670 RETURN
2680 REM=====================
2690 REM affichage d'un objet
2700 REM=====================
2710 LOCATE#3,1,1
2720 FOR ppoint=0 TO 143
2730 IF pointc(objet,ppoint)=allume THEN PRINT#2,CHR$(240);ELSE PRINT#2," ";
2740 NEXT ppoint
2750 RETURN
2760 REM=============================================================
2770 REM effacement de l'affichage graphique et du tableau correspondant
2780 REM=============================================================
2790 INPUT "Pour continuer, appuyer sur <ENTER>",rep$
2800 FOR x=1 TO 12
2810  FOR y=1 TO 12
2820    ppoint=14*y+x
2830    point(ppoint)=eteint
2840  NEXT y
2850 NEXT x
2860 CLS:CLS#2:CLS#3
2870 RETURN
2880 REM==============================
2890 REM sauvegarde de la base d'objets
2900 REM==============================
2910 IF nobjets=0 THEN PRINT "Impossible:pas d'objets en memoire":RETURN
2920 INPUT "Nom de fichier";nom$
2930 OPENOUT nom$
2940 PRINT#9,nobjets
2950 FOR objet=0 TO nobjets-1
2960   PRINT#9,nom$(objet)
2970   PRINT#9,ntrous(objet)
2980   PRINT#9,npointes(objet)
2990   PRINT#9,nfigures(objet)
3000   PRINT#9,nangles(objet)
3010   PRINT#9,gx(objet)
3020   PRINT#9,gy(objet)
3030   FOR ppoint=0 TO 143
3040     PRINT#9,pointc(objet,ppoint)
3050   NEXT ppoint
3060 NEXT objet
3070 CLOSEOUT
3080 GOTO 730
3090 REM==========================
3100 REM lecture d'une base d'objets
3110 REM==========================
3120 INPUT "Nom de fichier";nom$
3130 OPENIN nom$
```

```
3140 INPUT#9,nobjets
3150 FOR objet=0 TO nobjets-1
3160   INPUT#9,nom$(objet)
3170   INPUT#9,ntrous(objet)
3180   INPUT#9,npointes(objet)
3190   INPUT#9,nfigures(objet)
3200   INPUT#9,nangles(objet)
3210   INPUT#9,gx(objet)
3220   INPUT#9,gy(objet)
3230   FOR ppoint=0 TO 143
3240     INPUT#9,pointc(objet,ppoint)
3250   NEXT ppoint
3260 NEXT objet
3270 CLOSEIN
3280 GOTO 730
```

CHAPITRE 2
RECONNAISSANCE DE LA PAROLE

QU'EST-CE QUE C'EST ?

Le reconnaissance de la parole (Rpa) est, avec la reconnaissance de formes, un des domaines les plus ouverts en matière d'intelligence artificielle. Pourquoi ? Tout simplement parce que les applications industrielles y sont aussi prometteuses que variées. Et qui dit industrie dit argent, nerf de la guerre en recherche théorique, tout spécialement aux US où le bénévolat est aussi rare que les subventions gouvernementales.

Un rapport du Stanford Research Institute (1979) estime que le marché de la parole devrait passer de $100 millions en 1984 à $250 millions en 1990. De plus, ce marché pourrait exploser dans la décennie suivante pour peu que le rapport performances/coût devienne suffisamment élevé.

On assisterait alors au même phénomène que celui que nous vivons avec la micro-électronique et à un degré moindre, avec la micro-informatique.

Autant dire que les investisseurs surveillent d'un regard avide ce marché et que les recherches s'y développent en quantité et en qualité.

Dans un contexte d'euphorie, les réussites foisonnent et nous en citerons quelques-unes parmi les plus spectaculaires.

Avant d'aborder les principaux domaines de recherche en reconnaissance de la parole, c'est-à-dire les principaux domaines d'applications industrielles il convient de revenir une fois de plus sur "l'esprit" qui fonde la reconnaissance de la parole.

Nous l'avons vu, la reconnaissance de la parole intervient typiquement au niveau PERCEPTION du processus de l'intelligence artificielle. Pourtant, en aucun cas ce terme de perception ne signifie "superficiel" : les mécanismes qui guide la reconnaissance de la parole chez l'homme sont parmi les plus complexes et les plus difficiles à imiter, au même titre que la reconnaissance d'images.

LES DOMAINES D'APPLICATIONS

Ils sont nombreux et prometteurs. Si la plupart restent cantonnés aux domaines de la recherche (probablement plus pour longtemps), certains ont fait récemment une entrée remarquée dans notre quotidien. Parmi les projets qui mûrissent doucement dans les laboratoires de recherche et ceux déjà opérationnels, citons quelques exemples :

* entrée de données (scientifiques, bancaires, etc.)
* identification de personne
* renseignements automatisés (téléphoniques ou autres)
* appareils domestiques répondant à la voix
* automobiles
* jeux électroniques
* robotique
* assistance aux handicapés

LES SYSTÈMES EXISTANTS

Les systèmes de reconnaissance de la parole disponibles aujourd'hui ont été pour la plupart mis au point soit par de grands groupes industriels (preuve s'il en était besoin de l'avenir de la reconnaissance de la parole), soit par les universités. Citons :

* HEARSAY :	premier système de reconnaissance de la parole continue
* DRAGON :	fondé sur le modèle de Markov (75)
* LPARS :	(MIT) reconnaisseur de phrases sans étape phonétique, décodeur statistique d'IBM
* HWIN :	très élaboré du point de vue des contraintes linguistiques
* MYRTILLE :	système français pour interrogation d'un standard automatique.
* ESOPE :	système français, dialogue homme-machine

Que le premier constructeur d'ordinateurs (IBM) éprouve le besoin d'être présent dans les créneaux de la reconnaissance de la parole doit être considéré comme un signe révélateur de l'avenir industriel de cette branche de l'intelligence artificielle.

L'ÉTENDUE DU PROBLÈME

La principale difficulté en reconnaissance de la parole réside dans la reconnaissance de la parole continue. En effet, reconnaître des mots isolés ne présente plus guère de problèmes, compte tenu des progrès récents et spectaculaires de la micro-électronique dans ce domaine. A telle enseigne que les applications industrielles de ce cas particulier de la reconnaissance de la parole foisonnent dans un grand nombre de secteurs. Citons pour mémoire celui de l'automobile avec ces nouveaux modèles désormais capables d'obéir à des ordres simples tels que : "OUVRE PORTE" ou "ALLUME PHARES"... Ces systèmes font essentiellement appel à l'électronique et très peu, voire pas du tout, à l'intelligence artificielle.

Aussi, n'en parlerons-nous pas. Pas plus que nous ne parlerons de la toute première étape qui intervient dans le processus de reconnaissance de la parole : le traitement du signal.

Cette étape, extrêmement complexe, fait appel à des disciplines comme l'électronique (saisie du son, filtrage, échantillonnage, etc.) et l'algorithmique (série de Fouriers, compression, etc.) mais très peu à l'heuristique. Elle n'a donc pas sa place dans un ouvrage traitant de l'intelligence artificielle.

En fait, on peut parfaitement admettre de faire de la reconnaissance de la parole sur un micro-ordinateur comme l'Amstrad sans disposer d'un organe de perception des sons : on entrera directement les "sons", c'est-à-dire les phonèmes au clavier si le programme est capable d'effectuer une analyse phonétique, ou des mots si les compétences du programme ne démarrent qu'à l'analyse syntaxique ou sémantique.

Par exemple on entrera cette suite de syllabes au clavier :

"il/è/bi/in/mon/am/stra/de"

Dans un premier temps, le programme devra reconnaître des mots contenus dans son lexique de base :

"il est bien mon Amstrad"

Puis il vérifiera la syntaxe de la phrase, avec retour en arrière en cas d'anomalie (syntaxe incorrecte). Le programme SYNTAX présenté plus loin traite plus particulièrement cet aspect du problème.

Ensuite, le programme pourra effectuer une analyse sémantique des mots constituant la phrase, pour en comprendre le sens. Le mot Amstrad fait par exemple référence à un des micro-ordinateurs les plus doués de sa génération.

Le programme SEMAN, également proposé dans cet ouvrage, aborde cet aspect de la reconnaissance de la parole. Enfin, un programme exhaustif de la reconnaissance de la parole pourra s'intéresser au contexte pour réinterpréter éventuellement le sens de la phrase.

Quant au logiciel PHRASE que vous trouverez également parmi les exemples d'applications de reconnaissance de la parole sur Amstrad, il analyse le sens de la question que vous lui posez et sélectionne la réponse dans une liste de phrases que vous aurez saisies préalablement.

LES ÉTAPES

Comme nous venons d'en avoir un aperçu, la parole peut être analysée à différents niveaux. Seul le processus complet, incluant tous les niveaux permet par une analyse exhaustive, d'espérer un résultat fiable, ou du moins pas trop mauvais.

On distingue cinq étapes :

Etapes	Exemple	Programme
Phonologique : reconnaissance des phonèmes	l/e/c/h/i/e/n/e/s/t/g/r/o/s → le/chi/in/è/gro	
Lexicale : reconnaissance des mots	le/chi/in/è/gro → (le)/(chien)/(est)/(gros)	
Syntaxique : reconnaissance des phrases	(le)/(chien)/(est)/(gros) → /le chien est gros/...	Syntax
Sémantique : reconnaissance du sens du message	chien <=> animal, 4 pattes,... gros <=> poids plus lourd que la normale	Seman Phrase
Contextuelle : reconnaissance du sens dans le contexte	/le chien est gros/c'est un bon fusil/ → dans ce contexte, chien ne peut être un animal : itération à l'étape précédente.	

LES STRATÉGIES

Comme pour la reconnaissance de formes, il existe deux grandes familles de méthodes de reconnaissance de la parole : les méthodes structurelles et les méthodes statistiques. Seuls les systèmes se réclamant des deux à la fois parviennent à un certain niveau d'efficacité. Pour notre part, nous allons uniquement nous intéresser aux méthodes structurelles : l'aspect statistique du problème, quoique très important, reste peu spectaculaire et se prête mal à la vulgarisation. D'autre part, l'approche statistique en intelligence artificielle varie peu d'un domaine à l'autre : ce que nous avons déjà dit dans le chapitre sur la reconnaissance de formes reste vrai ici. Signalons simplement qu'une des méthodes statistiques (utilisée dans le décodeur statistique linguistique d'IBM) consiste à rechercher dans un lexique de mots, celui dont le signal (un son peut toujours être représenté sous forme de signal) se rapproche le plus du signal émis par un locuteur.

Le problème posé par la reconnaissance de la parole s'inscrit exactement dans la catégorie des problèmes d'intelligence artificielle. Qu'on en juge plutôt en énumérant ses composantes, que l'on retrouve en reconnaissance de formes, dans les systèmes experts, etc.

* Les données (phonèmes) sont incomplètes et erronées car les locuteurs sont des hommes faillibles.

Ex. : "t'as d'beaux zieux t'sais" au lieu de "tu as de beaux yeux tu sais".

* Les choix sont non déterministes : tout choix effectué à une étape donnée peut être remis en cause par une impossibilité à l'étape suivante. (erreur → apprentissage → retour arrière).

Exemple d'erreur phonétique détectée à l'étape sémantique (dialogue à lire avec l'accent vosgien) :

"– MirzA, elle a fait des PEUtits
– dèCHA ? (déjà)
– non des chiens"

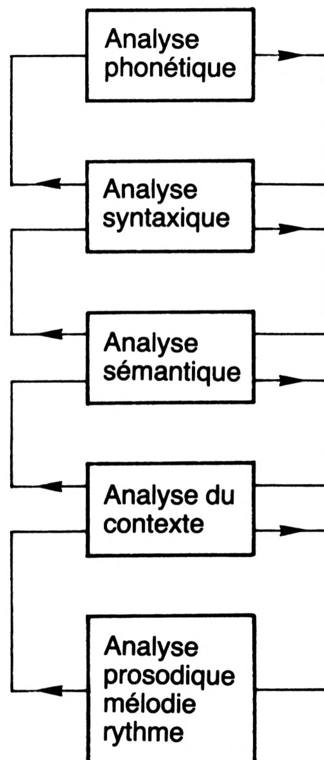
* Le problème est du type combinatoire : compte tenu du droit à l'erreur du locuteur, une suite de phonèmes peut être combinée d'un très grand nombre de façons. De même, dans l'étape syntaxique, les phrases valides sont en nombre très élevé.

Compte tenu de tous ces points, il est très intéressant d'aborder le problème de la reconnaissance de la signification d'un message au travers un modèle heuristique utilisant, comme pour les systèmes experts, une base de faits et une base de règles.

Ce modèle, comprenant au moins les cinq étapes que nous avons citées, consistera en une stratégie de recherche de la solution optimale (pas nécessairement unique). Cette stratégie reposera sur une combinaison des différentes étapes (analyse phonétique, syntaxique, etc.), avec retours en arrière en cas d'erreur et itérations successives pour affiner la résolution trouvée.

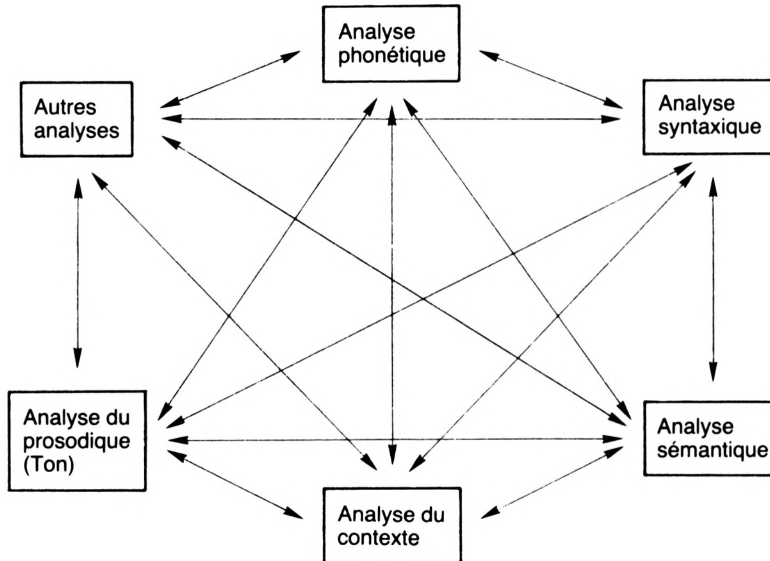
Le modèle le plus simple (sinon le plus plus efficace) est le modèle hiérarchique qui intègre la structure hiérarchique du message à reconnaître (phonèmes → mots → phrases → message).

Modèle hiérarchique



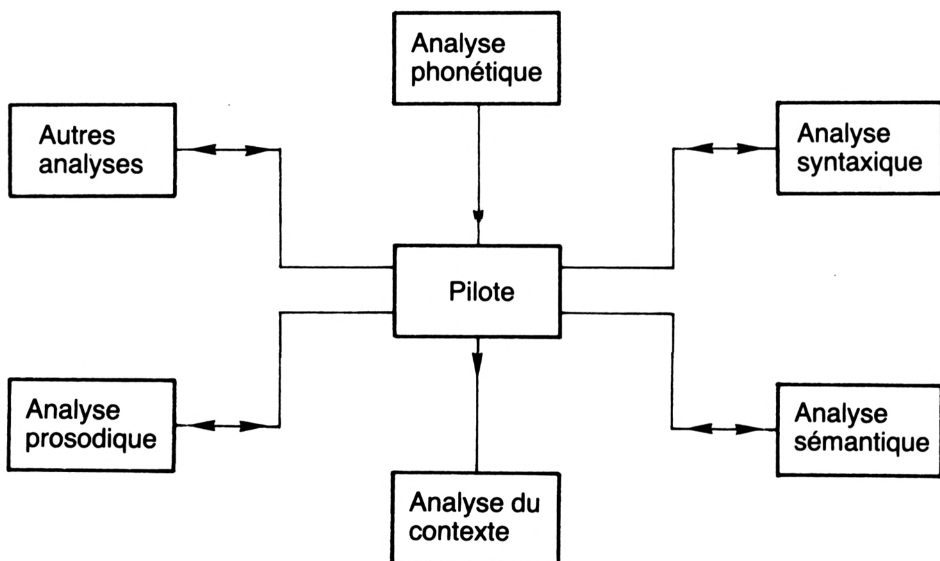
La reconnaissance se fait toujours sur deux niveaux adjacents à la fois, ce qui facilite les retours en arrière d'un niveau. Les étapes dont donc exécutées deux à deux : phonétique et syntaxique, puis syntaxique et sémantique, etc.

Le modèle multi-hiérarchique généralise l'approche précédente : toutes les étapes sont combinées deux à deux.



Modèle multi-hiérarchique

Le modèle distribué est piloté par une boîte programmable : selon le type de phrase, le locuteur, le contexte, la stratégie peut être adaptée en agissant sur ce "pilote". Ce modèle fournit donc plus de souplesse que les deux précédents.



Modèle pilote

L'ÉTAPE PHONÉTIQUE ET LEXICALE

Il s'agit en fait d'une double analyse : phonétique et lexicale. Pour chaque phonème reçu par le système (par micro, clavier, etc.), une hypothèse est faite sur son orthographe ainsi que sur les liens qu'il possède avec les précédents phonèmes. Une recherche est effectuée dans un lexique pour tenter de rapprocher la suite de phonèmes d'un mot connu. Pour ne pas surcharger le lexique avec toutes les orthographes et prononciations différentes d'un mot (ex. : animal, animaux), on cherchera à établir des règles d'accord, de conjugaison, etc.

règle 1 : <nom> + <suffixe al> + <pluriel> → <nom> + <suffixe aux>
règle 2 : <adj> + <suffixe eau> + <féminin> → <adj> + <suffixe elle>
règle 3 : <je> + <verbe> + <suffixe er> → <verbe> + <e>
: <tu> + <verbe> + <suffixe er> → <verbe> + <es>
: <il> + <verbe> + <suffixe er> → <verbe> + <e>
: <nous> + <verbe> + <suffixe er> → <verbe> + <ons>
: <vous> + <verbe> + <suffixe er> → <verbe> + <ez>
: <ils> + <verbe> + <suffixe er> → <verbe> + <ent>

Tous les mots de base n'entrant pas dans le cadre de ces règles simples devront être mis dans le lexique avec leurs variantes et leurs conjugaisons (pour les verbes). Ainsi le verbe être, irrégulier et donc trop isolé pour justifier une règle particulière à lui tout seul, sera mis dans le lexique sous les formes (être, suis, es, est, sommes, êtes, sont). On voit bien là l'intérêt de quelques règles lexicales : on parvient à diviser d'un facteur cinq à dix la taille du lexique.

L'ÉTAPE SYNTAXIQUE

Il s'agit là, à partir des mots identifiés pendant l'étape lexicale, de construire des phrases correctes du point de vue de la syntaxe. Ce qui signifie que cette étape s'occupe de vérifier que la suite des mots constitue bien une phrase respectant des règles de grammaire prédéfinies. La reconnaissance de la parole étant non déterministe (des erreurs peuvent se glisser dans la prononciation, etc.), cette étape va permettre de "remettre les pendules à l'heure" et d'éliminer parmi la liste de mots reconnus par l'analyse lexicale, ceux qui n'entrent pas dans le cadre d'une phrase reconnue comme telle.

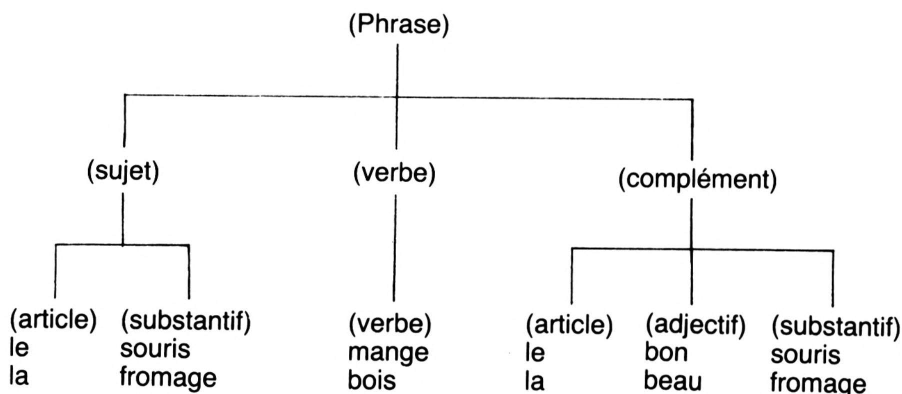
C'est donc la grammaire de l'analyseur syntaxique qui décidera de la validité de telle ou telle construction. En cas d'anomalie, deux hypothèses seront à prendre en considération : soit une erreur est apparue dans l'étape d'analyse phonétique, soit le locuteur a pris quelques libertés avec la grammaire. Pour choisir entre ces deux hypothèses, il faudra d'une part itérer l'étape précédente, d'autre part attendre l'étape sémantique. Toute la reconnaissance de la parole va consister (et c'est là l'objet d'une stratégie de la reconnaissance) à effectuer des allées et venues entre l'étape syntaxique et lexicale, entre l'étape sémantique et syntaxique.

Il existe trois types de grammaire : à états finis, à contexte libre et langage naturel. Le premier type s'adresse à des langages rudimentaires, par ex. : les langages de commande de machines (robots, machines à commandes numériques, etc.). Le second type concerne tous les langages artificiels (BASIC, COBOL, PASCAL, etc.). Le troisième type est la grammaire bien connue des écoliers.

On décrit une grammaire sous forme de règles. Voici un exemple très simple de grammaire.

Règle 1 : <phrase> → <sujet> <verbe> <complément>
Règle 2 : <sujet> → <article> <substantif>
Règle 3 : <complément> → <article> <adjectif> <substantif>
Règle 4 : <substantif> → <fromage> <souris>
Règle 5 : <article> → <le> <la>
Règle 7 : <verbe> → <mange> <boit>

Il est possible de représenter cette grammaire sous une forme plus imagée.



Une phrase correcte du point de vue de la syntaxe sera :

LA SOURIS MANGE LE BEAU FROMAGE

Mais celle-ci le sera aussi :

LA SOURIS BOIT LE BEAU FROMAGE

ou encore

LE FROMAGE MANGE LA BELLE SOURIS
(on suppose que les règles d'accord font partie de la grammaire)

Ainsi, la syntaxe ne permet pas d'éliminer les phrases correctes du point de vue de la syntaxe, mais qui n'ont aucun sens. Il faut pour cela faire intervenir l'étape suivante : l'analyse sémantique.

L'ÉTAPE SÉMANTIQUE

Le rôle de la sémantique est de relier les symboles (mots) aux concepts (significations). Par exemple, on pourra avoir les concepts de JOIE, TRISTESSE, INDIFFÉRENCE. On reliera les mots "rire, jouer, bruit, gai" à la JOIE. Puis les mots "pleurer, mort, larme, triste" au concept de TRISTESSE. Enfin tous les autres mots du lexique au concept d'INDIFFÉRENCE.

Avec ces éléments très rudimentaires, on sera capable d'estimer si une phrase exprime l'un ou l'autre des sentiments désignés. Par exemple :
"Je ris et je suis gai" fait référence à la JOIE.
"Son père est mort et il pleure" fait référence à la TRISTESSE.
"Il fait beau ce matin" fait référence à l'INDIFFÉRENCE.

Certaines contraintes sémantiques seront introduites pour aider à la reconnaissance. Par exemple une phrase ne pourra pas faire référence à plusieurs des trois concepts à la fois (TRISTESSE et JOIE, etc.). Une phrase comme "je ris et je pleure" sera considérée comme sémantiquement incorrecte avec ces hypothèses.

Pour conclure, disons que les systèmes utilisant simultanément des contraintes sémantiques et syntaxiques sont encore très rares.

LES AUTRES ÉTAPES

Les deux autres étapes qui permettent d'affiner les résultats obtenus à ce stade sont : l'analyse du contexte et la prosodie.

La première permet de relier les phrases entre elles pour faire ressortir le sens général du discours. Cela peut permettre de finir les phrases incomplètes ou de tenir pour négligeables celles qui jurent avec le reste du discours.

Quant à la prosodie, elle consiste à interpréter le rythme, l'intensité et la mélodie de la phrase. Cette analyse permet de déterminer si une phrase est affirmative, interrogative, etc. ainsi que des sentiments : la joie, la colère,... Ce dernier point peut donc permettre de vérifier certaines conclusions de l'analyse sémantique. Hélas, mille fois hélas, il est difficile, voire impossible de disposer sur un micro comme l'Amstrad des informations nécessaires à cette analyse.

En guise de conclusion, disons que pour l'instant, il n'existe pas de systèmes élaborés tenant compte de tous les facteurs cités : nous ne doutons pas que c'est là la voie de l'avenir et que ce type de système, à analyses multiples, ne manquera pas bientôt de faire une entrée remarquée.

PRÉSENTATION DU PROGRAMME SYNTAX GÉNÉRATEUR DE TEXTE

COMMENT L'UTILISER ?

Ce programme génère des phrases en français, syntaxiquement et sémantiquement correctes. Ces phrases sont fabriquées à partir de "phrases arborescentes", créées par l'utilisateur et stockées dans des lignes DATA sous forme d'expressions contenant des chaînes de caractères, des parenthèses et des opérateurs (un exemple figure dans le programme de la ligne 740 à 800). Le logiciel évalue chaque expression (on pourrait aussi dire qu'il choisit un chemin dans chaque arbre), ce qui mène à la création d'une phrase. Les opérateurs sont :

– la concaténation, notée +, qui mène, comme son nom l'indique, à la concaténation de deux ou plusieurs chaînes de caractères.

– le “ou aléatoire”, qui, appliqué à une série de chaînes de caractères, conduit le programme à choisir aléatoirement l'une d'entre elles.

Ainsi l'évaluation de l'expression :
(il est+(indispensable * primordial)+de+(manger * boire))

Pourra mener à quatre résultats différents :
il est indispensable de manger
il est indispensable de boire
il est primordial de manger
il est primordial de boire

Les copies d'écran 1 et 2 illustrent ce mécanisme sur un exemple plus complexe.

Les phrases arborescentes données en DATA ne sont bien sûr que des exemples, et peuvent être modifiées à votre guise, il vous faudra alors suivre quelques règles :

– il n'y a pas d'opération prioritaire. Des parenthèses bien placées sont donc indispensables pour lever toute ambiguïté. Ainsi l'expression :
(ce chat est+gris * tigré)
est incorrecte car le programme ne sait pas laquelle des deux opérations (+ ou *), il doit effectuer en premier. Il aurait fallu écrire par exemple :
(ce chat est+(gris * tigré))

– l'ensemble de l'expression doit être entouré de deux parenthèses, la dernière refermant la première :
(ce chat * ce chien)+(est blanc * est noir)
est incorrecte car la seconde parenthèse referme la première, et la dernière referme l'avant-dernière. Il aurait fallu écrire :
((ce chat * ce chien)+(est blanc * est noir))

– d'une manière générale, à toute parenthèse ouvrante doit correspondre une parenthèse fermante (comme dans une expression algébrique).

Plus vous imbriquerez les niveaux de parenthèses et plus le nombre de phrases possibles sera élevé, on aboutit même rapidement à une véritable explosion combinatoire.

Ce programme est inclus dans le programme “analyse psychologique” qui contient un grand nombre de phrases arborescentes.

COMMENT ÇA MARCHE ?

Ce programme saisit les arbres stockés en DATAs sous forme d'expressions alphanumériques et en extrait aléatoirement un chemin qui part du nœud initial et se termine à un des nœuds terminaux.

Ce qui revient à évaluer l'expression alphanumérique composée de chaînes de caractères, de parenthèses et d'opérateurs (concaténation notée + et “ou aléatoire” notée *). Ce genre d'expression n'est pas très différent des expressions algébriques de type basic ou pascal, seule la nature des opératiions et des données changent.

L'évaluation d'une expression s'effectue par l'intermédiaire de trois piles :

– une pile de données (qui contiendra donc des chaînes de caractères) ;
– une pile d'opérateurs (+ et *) ;
– une pile dite exécution dans laquelle les données sont empilées en attente d'exécution d'un opérateur, l'opération "ou aléatoire" ne peut en effet être appliquée que globalement à plusieurs chaînes de caractères (et non par évaluation deux à deux) de façon à conserver l'équiprobabilité du tirage de chacune d'entre elle. Ainsi l'expression (A*B*C*D) provoquera l'empilage dans cette pile de A, B, C et D puis le choix aléatoire (opérateur *) de l'une des quatre lettres.

Voyons le processus global d'évaluation. L'analyse d'une expression est effectuée de gauche à droite selon les règles suivantes :

– si une chaîne de caractères est rencontrée, elle est empilée dans la pile de données ;
– si un opérateur ou une parenthèse ouvrante sont rencontrés, ils sont empilés dans la pile opérateur ;
– si une parenthèse fermante est rencontrée, les chaînes de la pile de données sont dépilées une à une, puis empilées dans la pile exécution, et les opérateurs (que l'on peut mettre au singulier puisqu'ils sont toujours identiques à l'intérieur d'un même niveau de parenthèses) sont dépilés au même rythme que les chaînes de la pile donnée. Ce processus se répète jusqu'à ce qu'une parenthèse ouvrante soit rencontrée sur la pile opérateurs. L'opérateur peut alors être appliqué aux données de la pile exécution que l'on vide immédiatement. Le résultat est placé sur la pile de données.

En remontant ainsi jusqu'à la première parenthèse ouvrante, la phrase est construite et se trouve sur la pile opérante. Il ne reste plus qu'à l'afficher.

Le processus d'évaluation que nous venons de décrire est, vous l'avez remarqué, assez complexe, l'analyse montre sur un exemple, les différents états de trois piles durant cette évaluation.

PRINCIPALES VARIABLES :

pdo$(i) pile de données
pop$(i) pile d'opérateurs
pex$(i) pile d'exécution

ÉVALUATION DE LA PHRASE ARBORESCENTE :

(LE CHAT EST + (NOIR * GRIS))

PDO:
POP: (
PEX:

1 : empilage de (

PDO : LE.CHAT.EST
POP : (
PEX :

2 : empilage de LE CHAT EST

PDO : LE.CHAT.EST
POP : (+
PEX :

3 : empilage de +

PDO : LE.CHAT.EST
POP : (+ (
PEX :

4 : empilage de (

PDO : LE.CHAT.EST NOIR
POP : (+ (
PEX :

5 : empilage de NOIR

PDO : LE.CHAT.EST NOIR
POP : (+ (*
PEX :

6 : empilage de *

PDO : LE.CHAT.EST NOIR GRIS
POP : (+ (*
PEX :

7 : empilage de GRIS

PDO : LE.CHAT.EST NOIR
POP : (+ (
PEX : GRIS

8 : GRIS va de PDO à PEX
et * est dépilée

PDO : LE.CHAT.EST
POP : (+
PEX : GRIS NOIR

9 : NOIR va de PDO à PEX
et (est dépilée

PDO : LE.CHAT.EST GRIS
POP : (+
PEX :

10 : * est appliqué à GRIS et NOIR, supposons que le résultat (aléatoire) soit GRIS, GRIS est alors placé sur PDO

PDO : LE.CHAT.EST
POP : (
PEX : GRIS

11 : GRIS va de PDO à PEX
+ est dépilé

PDO :
POP :
PEX : GRIS LE.CHAT.EST

12 : (est dépilée
LE.CHAT.EST va de PDO à PEX

PDO : LE.CHAT.EST.GRIS
POP :
PEX :

13 : + est appliqué à LE.CHAT.EST et GRIS
le résultat final (POP et PEX étant vides), est placé sur PDO

PDO est la pile de données
POP est la pile opérateur
PEX est la pile exécution

Pour éviter toute confusion, les différents mots d'une même donnée sont reliés par des points.

```
G E N E R A T E U R   D E   T E X T E   S Y N T A X I Q U E M E N T
C O R R E C T   E T   S E M A N T I Q U E M E N T   C O H E R E N T

D'aucuns affirment sans craindre une critique moqueuse
que l'intelligence electronique est
a notre cerveau
ce que le chemin de fer est au cheval .
Pourtant , train ou machine a vapeur constituent des prolongements plus ou moins
 subtils
de nos muscles rouges .
Le traitement symbolique s'en prend directement aux arcanes du raisonnement

Pour recommencer, appuyer sur <ENTER>

Certains pensent
que l'intelligence artificielle est
 a la matiere grise
ce que la mecanisation est au travail manuel .
Pourtant , chemin de fer ou machine a vapeur constituent des substituts
de nos moyens physiques .
L'IA s'en prend directement aux principes de la pensee

Pour recommencer, appuyer sur <ENTER>
```

Exemple 1 d'exécution de SYNTAX

```
G E N E R A T E U R   D E   T E X T E   S Y N T A X I Q U E M E N T

C O R R E C T   E T   S E M A N T I Q U E M E N T   C O H E R E N T


D'aucuns affirment sans craindre une critique moqueuse
que l'intelligence artificielle est
au cerveau humain
ce que la mecanisation est a la force musculaire .
Pourtant , chemin de fer ou machine a vapeur constituent des substituts
de nos capacites physiques .
L'IA simule etroitement les fonctions cerebrales

Pour recommencer, appuyer sur <ENTER>

Certains pensent
que l'intelligence electronique est
a notre cerveau
ce que le chemin de fer est au cheval .
Pourtant , chemin de fer ou chevaux vapeur sont des substituts
de nos capacites physiques .
Le traitement symbolique simule etroitement les fonctions cerebrales

Pour recommencer, appuyer sur <ENTER>
```

Exemple 2 d'exécution de SYNTAX

```
10 REM********************
20 REM  SYNTAX (T. L.-A.)
30 REM********************
40 REM---initialisation---
50 MODE 2:CLS
60 DEFINT a-z
70 DIM pdo$(30),pop$(30),pex$(20)
80 PRINT "G E N E R A T E U R   D E   T E X T E   S Y N T A X I Q U E M E N T"
90 PRINT
100 PRINT "C O R R E C T   E T   S E M A N T I Q U E M E N T   C O H E R E N T"
110 PRINT:PRINT
120 REM---LECTURE D'UNE PHRASE ARBORESCENTE---
130 READ CH$
140 IF CH$="FIN" THEN PRINT:INPUT "Pour recommencer, appuyer sur <ENTER>",rep:PR
INT:RESTORE:GOTO 130
150 d=0:o=0:e=0
160 REM---RECHERCHE D'ERREURS DE PARENTHESES---
```

```
170 FOR I=1 TO LEN(CH$)
180    IF MID$(CH$,I,1)="(" THEN I1=I1+1:GOTO 200
190    IF MID$(CH$,I,1)=")" THEN I2=I2+1
200 NEXT I
210 IF I1<>I2 THEN PRINT "Erreur de parenthese":END
220 REM----------------------------------
230 REM Creation d'une phrase aleatoire
240 REM a partir d'une phrase arborescente
250 REM----------------------------------
260 FOR i=1 TO LEN(ch$)
270 car$=MID$(ch$,i,1)
280 IF car$=")" THEN 350
290 IF car$<>"(" AND car$<>"+" AND car$<>"*" THEN 330
300 IF mot$<>"" THEN pdo$(d)=mot$+" ":mot$="":d=d+1
310 pop$(o)=car$:o=o+1
320 GOTO 610
330 mot$=mot$+car$
340 GOTO 610
350 IF mot$<>"" THEN pdo$(d)=mot$+" ":mot$="":d=d+1
360 GOSUB 690
370 op$=pop$(o-1):o=o-1
380 GOSUB 690
390 IF pop$(o-1)=op$ THEN o=o-1:GOTO 380
400 IF pop$(o-1)<>"(" THEN PRINT "Erreur de parenthese":END
410 o=o-1
420 IF op$="*" THEN 580
430 REM-----------------------------------
440 REM  execution de l'operateur "AJOUTER"
450 REM              note +
460 REM-----------------------------------
470 pdo$(d)=""
480 FOR J=e-1 TO 0 STEP -1
490    pdo$(d)=pdo$(d)+pex$(j)
500 NEXT j
510 d=d+1
520 e=0:REM vidage de la pile execution
530 GOTO 610
540 REM----------------------------------------
550 REM  execution de l'operateur "OU ALEATOIRE"
560 REM              note *
570 REM----------------------------------------
580 pdo$(d)=pex$(INT(RND*E))
590 d=d+1
600 e=0
610 NEXT i
620 REM---affichage de la phrase aleatoire---
630 PRINT pdo$(0)
640 GOTO 130
```

```
650 REM---------------------------------
660 REM Depilage de la pile de donnees et
670 REM  empilage dans la pile execution
680 REM---------------------------------
690 pex$(e)=pdo$(d-1):e=e+1:d=d-1
700 RETURN
710 REM--------------------------------
720 REM Reservoir de phrases aleatoires
730 REM--------------------------------
740 DATA "((Certains*Certaines sommites scientifiques*D'aucuns)+((affirment+(ave
c conviction*(sans craindre une critique+(moqueuse*souvent agressive))))*(preten
dent+(tranquillement*sans complexe))*pensent*estiment))"
750 DATA "(que l'intelligence+(artificielle*electronique)+est)"
760 DATA "((a notre cerveau*(au cerveau+(humain*(de+(l'homme*l'homo sapiens)))))
* a la matiere grise)"
770 DATA "(ce que+(((la machine a vapeur*la mecanisation)+est+(au muscle*a la fo
rce musculaire*au travail manuel))*le chemin de fer est au cheval)+.)"
780 DATA "((Mais*Pourtant*Cela dit)+,+((train*chemin de fer)+ou+((machine a*chev
aux)+vapeur))+(sont*constituent*representent)+(des+(extensions*(prolongements+(p
lus ou moins+(sophistiques*subtils*raffines)))*substituts)))"
790 DATA "(de nos+(((moyens*possibilites*capacites)+physiques)*muscles rouges)+.
)"
800 DATA "((L'intelligence artificielle*L'IA*Le traitement symbolique)+((s'en pr
end directement aux+(arcanes*principes)+(du raisonnement*de la pensee))*(simule
etroitement+(les fonctions cerebrales*(les comportements+(mentaux*humains))))))"

810 DATA "FIN"
```

PRÉSENTATION DU PROGRAMME SEMAN (DIALOGUE ET ANALYSE PSYCHOLOGIQUE)

COMMENT L'UTILISER ?

Il est inutile d'être un spécialiste pour utiliser ce logiciel, il suffit de se plier à cette seule règle, au demeurant peu contraignante :

Lorsque le message "VOUS:" apparaît, c'est à vous de taper "ce que vous voulez", l'ordinateur vous répondra et un dialogue riche et varié s'établira entre vous et l'ordinateur. Le nombre de réponses possibles de la part de l'ordinateur est quasiment infini, mais des pans entiers du programme ne seront peut-être jamais utilisés si vous ne savez pas y accéder par une conduite intelligente du dialogue. Ce logiciel s'explore en effet un peu à la façon d'un jeu d'aventure et vous fera passer, nous l'espérons, de longues heures de détente.

Pour votre gouverne, sachez que ce logiciel reconnaît de nombreux "gros" mots, présents sous forme codée dans le listing (censure oblige !) et détectera la plupart du temps les phrases incohérentes. Il vous préparera alors une réponse "salée".

Il nous faut maintenant parler de la face cachée de ce logiciel : il vous espionne et met en permanence à jour un tableau dont chaque élément mesure un des aspects de votre personnalité. A tout moment vous pouvez tout savoir sur celle-ci en tapant le mot "bilan". Sachez toutefois que ce bilan sera d'autant plus précis et détaillé que vous aurez conversé avec des phrases longues et sensées.

```
A tout moment, vous pouvez avoir un bilan complet
en tapant BILAN
ORDINATEUR: bonjour! parlez-moi de vous
VOUS:? j'aime la vie et je suis joyeux
ORDINATEUR:cela a au moins le merite d'etre positif
VOUS:? et vous
ORDINATEUR:nous sommes la pour discuter de vous
VOUS:? je crois que je suis sympathique
ORDINATEUR:parlez-moi de vos centres d'interets
VOUS:? j'aime l'informatique
ORDINATEUR:pour vous est-ce un sentiment eternel ?
VOUS:? un sentiment ... pour les ordinateurs ...
ORDINATEUR:pour vous est-ce un sentiment incomprehensible ?
VOUS:? l'informatique ne m'inspire que de l'interet
ORDINATEUR:voila un domaine qui nous passionne tous les deux
VOUS:? je te trouve reellement idiot
ORDINATEUR:gardons notre calme
VOUS:? J'insiste, tu es bete
ORDINATEUR:Oh moi,vous savez...
VOUS:? hum
ORDINATEUR:enchainez ...
VOUS:? il fait chaud et je suis fatigue
ORDINATEUR:etes-vous la pour discuter du temps?
VOUS:? oui et non
ORDINATEUR:que pensez-vous des OVNIS
VOUS:? les OVNIS c'est du bidon
ORDINATEUR:parlez-moi de votre personnalite
VOUS:? je suis intelligent
ORDINATEUR:parlez-moi de vous
VOUS:? je suis fatigue mais optimiste
ORDINATEUR:ca ira mieux le moins prochain
VOUS:? et le travail ... ah! le travail !
ORDINATEUR:pourquoi ne pas creer votre propre entreprise ?
VOUS:? il faut de l'argent
ORDINATEUR:l'argent ne fait pas le bonheur
VOUS:? ben des fois ca aide ... a manger par exemple
ORDINATEUR:faites-vous la cuisine?
VOUS:? la cuisine ? bof
ORDINATEUR:que pensez-vous des OVNIS
```

```
VOUS:? cette question vous obsede
ORDINATEUR:nous sommes la pour discuter de vous
VOUS:? uyjhtgfrewdsuyjewdhgsf
ORDINATEUR:mon clavier ne vous a rien fait
VOUS:? uirehjgfsdiujrfediujrfed
ORDINATEUR:elle est bien bonne!et celle la,vous la connaissez : pordkfe
VOUS:? bilan
joie: 23
insouciance: 5
optimisme: 18
egocentrisme: 20
amour: 40
avarice: 20
positivisme: 4
agressivite: 10
VOUS:? moi agressif ? tu m'enerves la !
ORDINATEUR:et la negociation?
VOUS:? bilan
joie: 23
insouciance: 5
optimisme: 18
egocentrisme: 22
amour: 37
avarice: 22
positivisme: 4
agressivite: 20
VOUS:? merde
ORDINATEUR:ne me poussez pas a bout
VOUS:? si
ORDINATEUR:je vous ecoute
```

COMMENT FONCTIONNE-T-IL ?

Le fonctionnement de ce logiciel passe par deux phases bien distinctes :

- l'analyse de la phrase tapée par l'utilisateur,
- la création d'une réponse.

L'analyse de la phrase tapée par l'utilisateur est purement "sémantique" et ne passe par aucune analyse syntaxique. Cette seconde solution a été écartée d'emblée car les logiciels fonctionnant ainsi se contentent la plupart du temps de reconjuguer les phrases tapées et conduisent rapidement à la lassitude de l'utilisateur.

Avant d'entrer dans l'analyse elle-même, précisons la structure des "connaissances" du logiciel, composée de six parties :

– une liste de “traits de caractères”, chacun étant accompagné de son contraire (lignes 650 à 690) ;

– une liste de mots-clefs associés à des sujets : voiture, informatique, temps, amour, etc. (lignes 740 à 870). En fait ces mots sont souvent de simples racines, ce qui augmente l'étendu du vocabulaire connu ;

– une liste de groupes de mots-clefs auquels sont associés des coefficients, précédés du numéro de trait de caractère auquel il se rapporte (lignes 150 à 600). Chaque groupe de mots est suivi des mots de sens opposé, leur coefficients étant inversés. Ces derniers n'apparaissent pas dans la liste, ils sont calculés après inversion durant la lecture des DATA ;

– une liste de phrases arborescentes (dont le développement, comme dans le programme “générateur de texte”, peut aboutir à des milliers de phrases différentes, mais toujours cohérentes). Chacune de ces phrases est associée à un trait de caractère (ou état d'esprit) ;

– une liste de phrases arborescentes, dont chacune est associée à un sujet abordé par l'interlocuteur humain ;

– une liste de phrases arborescentes associées aux cas particuliers : relance lorsque le dialogue est trop neutre, réponse lorsqu'une phrase est trop courte, ou encore quand elle est incohérente.

L'analyse de la phrase tapée par l'utilisateur

Pour expliquer son fonctionnement, nous allons nous baser sur un exemple. Imaginons que l'utilisateur ait tapé la phrase :

“Je suis content car l'avenir s'annonce merveilleux”

Examinons maintenant chacun des mots qui la compose :

“je” possède un coefficient de 5 pour l'égocentrisme
“content” possède un coefficient de 10 pour la joie, 3 pour l'optimisme, 5 pour le positivisme.
“avenir” possède un coefficient de 10 pour l'optimisme, 5 pour la joie, 5 pour l'insouciance et 5 pour le positivisme.
“merveilleux” possède un coefficient de 10 pour le bonheur, 8 pour la joie, 6 pour l'insouciance et 5 pour l'optimisme.
les autres mots ne font pas partie de la liste des mots-clefs.

Après avoir repéré des mots, le programme effectue le score obtenu par chaque trait de caractère (ligne 2440 à 2550).

égocentrisme totalise 5 points
optimisme totalise 3 + 10 + 5 = 18 points
positivisme totalise 5 + 5 = 10 points
insouciance totalise 5 + 6 = 11 points
bonheur totalise 10 points
joie totalise 10 + 5 + 8 = 23 points

La création d'une réponse

Parmi ces six traits de caractères qui obtiennent une note non nulle, l'un est choisi aléatoirement et c'est à partir des phrases arborescentes qui lui sont associées que le programme va construire une réponse, grâce au "générateur de texte". Le hasard intervient à deux niveaux, il y a donc deux raisons pour qu'à phrases égales, le programme génère deux réponses différentes. D'autre part, la note générale (c'est-à-dire depuis le début du dialogue) de chaque trait de caractère est augmentée du score obtenu par la phrase. Ce sont ces scores qui sont exploités par l'ordre "bilan".

Dans le cas ou aucun des mots de la phrase ne font partie de la liste des mots-clefs associés aux traits de caractère, le programme cherche à détecter des mots-clefs associés à des sujets traités (lignes 740 à 870), un des sujets abordés est choisi aléatoirement et provoque la génération d'une réponse à partir des phrases arborescentes correspondant à ce sujet.

Si des mots-clefs des deux types sont détectés, le programme orientera aléatoirement la recherche d'une réponse vers l'une ou l'autre des méthodes.

Enfin, quelques cas particuliers peuvent se présenter :

– l'utilisateur a tapé "n'importe quoi", par simple curiosité, en plaquant sa main sur le clavier. Ceci provoque généralement la frappe d'une chaîne de caractères, plus longue que la plupart des mots français. Si elle dépasse quinze caractères, cela entraîne la génération d'une réponse créée par la phrase arborescente ABSURDE$ (ligne 1450) ;

– l'utilisateur a tapé une phrase trop courte (moins de sept caractères). Une réponse est alors construite à partir de la phrase arborescente TROP.COURT$ (ligne 1440) ;

– l'utilisateur a tapé une phrase qui ne contient aucun mot clef connu, une réponse destinée à relancer le dialogue est générée à partir de la phrase arborescente RELANCE$ (ligne 1430).

LISTE DES PRINCIPALES VARIABLES

mot$(i) tableau des mots ou racines de mots associés à des traits de caractère ;
mot2$(i) tableau des mots associés à des "sujets de conversation" ;
qual$(i, j) chaînes de caractères désignant les qualités (j = 0) et leur contraire (j = 1) ;
niveau(i,j) importance sémantique du mot i pour la qualité j ;
caractère(i) score atteint par l'utilisateur pour la qualité i ;
sujet(i) sujet de conversation qu'implique le mot i ;
phpcar$(i,j) phrases potentielles i associées à la qualité j = 0 ou à la qualité inverse (donc le défaut), j=1 ;
phsuj$(i) phrases potentielles associées au sujet i ;

pdo$(i) pile de données
pop$(i) pile opérateurs
pex$(i) pile exécution

} ces 3 tableaux interviennent dans le générateur de texte.

```
10 REM *****************
20 REM   SEMAN (T. L.-A.)
30 REM *****************
40 MODE 2
50 DEFINT a-z
60 DIM pdo$(30),pop$(30),pex$(20)
70 nmots=180:nmots2=150:nqual=15:nsujets=14
80 DIM qual$(nqual,1),niveau(nqual,nmots),sujet(nmots2),caractere(nqual)
90 DIM buff1$(30),buff2$(30),buff3(20)
100 DIM mot$(nmots),mot2$(nmots2),absurde$(70)
110 DIM phpcar$(nqual,1),phpsuj$(nsujets)
120 REM========================================
130 REM mots associes a des traits de caractere
140 REM========================================
150 DATA trist,pleur,larm,f1
160 DATA joy,joi,ris,rir,souri,fete,marre,foire,f2
170 DATA 1,10, 3,8, 10,5, f3
180 DATA stress,angoiss,deprim,cafard,bourdon,desesp,souci,probleme,f1
190 DATA insoucian,f2
200 DATA 2,10, 3,5, 10,5, f3
210 DATA pessimi,passe,f1
220 DATA optimi,avenir,futur,f2
230 DATA 3,10, 1,5, 2,5, 10,5, f3
240 DATA mechan,vilain,f1
250 DATA genti,amical,sympatique,sage,f2
260 DATA 4,10, 6,2, 10,5, 12,5, f3
270 DATA " ego"," moi "," je "," j'"," mon "," ma "," mes ",f1
280 DATA altruis,autre,f2
290 DATA 5,2, 9,2, f3
300 DATA hai,detest,f1
310 DATA amour,aime,sentiment,f2
320 DATA 6,10, 1,2, f3
330 DATA "zz xlm ",zznviwv,zzhzol,zzhzozfw,zzulfgi,zzvmxfov,zzgzxsv,zzkfgzrm,zzk
fgv,f1
340 DATA s'il vous plait,merci,bonjour,salut,f2
350 DATA 7,10, 4,8, f3
360 DATA sport,natat,foot,jogg,course,fort,musc,tennis,ski,f1
370 DATA seden,gros,gras," mou",f2
380 DATA 8, 10, f3
390 DATA avar,gard,f1
400 DATA genereu,partage," don",offr,offert,f2
410 DATA 9,10, 5,8, 4,2, f3
420 DATA " negat "," pas "," non "," ne "," n'"," rien ",contr," mais ",f1
430 DATA positi," oui "," pour ",f2
440 DATA 10,2,f3
450 DATA malheur,f1
460 DATA bonheur,heureux,merveilleux,f2
470 DATA 11,10, 1,8, 2,6, 3,5, f3
480 DATA " peur",effraye,secour,f1
```

```
490 DATA courage,temeraire,f2
500 DATA 12,10,f3
510 DATA " fou "," fol",debile,cingle," tare",idiot,cretin,f1
520 DATA f2
530 DATA 15,10,f3
540 DATA zzhvcv,zzerlo,zzuifhgi,zzyrgv,zzkrmv,zzxlfroov,zzxszggv
550 DATA zzhvrm,zzmfv,zzuvnnv,zzyzrhv,f1
560 DATA f2
570 DATA 14,10,f3
580 DATA agress,mordre," bat",guerre,f1
590 DATA paix,calme,passifis,f2
600 DATA 15,10,6,3,f3
610 DATA fin
620 REM=========================================
630 REM mots representant les traits de caractere
640 REM=========================================
650 DATA 1,tristesse,joie,2,deprime,insouciance,3,pessimisme,optimisme
660 DATA 4,mechancete,gentillesse,5,egocentrisme,altruisme,6,haine,amour
670 DATA 7,grossierete,politesse,8,sportivite,sedentarite,9,avarice,generosite
680 DATA 10,negativisme,positivisme,11,malheur,bonheur,12,peur,courage
690 DATA 13,folie,sainete d'esprit,14,libydo,libydo,15,agressivite,calme
700 DATA f1
710 REM==============================
720 REM mots associes aux sujets types
730 REM==============================
740 DATA 1,voiture,bagnole,route,f1
750 DATA 2,ordinateur,micro,informatiq,disquette,f1
760 DATA 3,pluie,soleil,chaud,froid,pluv,humid,nuage,neige," gel",f1
770 DATA 4,amour,aime,coeur,f1
780 DATA 5,metaphysique," ame",univers,creation," vie ",f1
790 DATA 6,dieu,deesse,relig," ange",paradi,enfer,catholi,chretien,arabe,juif,f1

800 DATA 7,travail,boulot,bosse,trime,f1
810 DATA 8," vous "," toi "," tu "," ton "," tes ",f1
820 DATA 9,annee,nostalgie,vieil,chomeur,chomage,boulot,f1
830 DATA 10,science,tech,chimi,math,astrono,sociolo,f1
840 DATA 11,sante,malad,bless,bobo,medecin,docteur,rhume,gripp,angine,fatigu,ner
f," las ",aie,ouille,f1
850 DATA 12,nourr,bouff,restau,repas," mange"," dine ",dejeun,faim,alim,viande,l
egume,f1
860 DATA 13,mort,morb,au-dela,cimetiere,tombe,patres,f1
870 DATA 14,argent,uirx,paye,franc,brique,billet,monnaie,lhvroov,kltmlm,f1
880 DATA f2
890 REM============================================
900 REM phrases arborescentes des reponses en rapport
910 REM avec les traits de caractere
920 REM============================================
930 DATA "(Pourquoi+(pleurez-vous*ces larmes)+,+(nostalgie*amour decu)+ou peut-e
tre parent perdu?)"
```

```
940 DATA "((super!*continuez!)+le vie est trop courte, il faut+(en profiter*en j
ouir au maximum))"
950 DATA "(((detendez*decrispez)+(vous*votre corps et votre esprit*votre ame))*(
d'ou vient cette angoisse?*depuis quand etes-vous stresse<e>?))"
960 DATA "((en tirez-vous parti*est-ce une qualite)+?)"
970 DATA "((L'avenir*le futur)+n'est+(peut-etre*sans doute)+(pas si+(sombre*laid
)))
980 DATA "((l'optimisme est une condition+(sine qua non de*indispensable a)+la r
eussite)*foncez! vous irez loin!)"
990 DATA "((Quel+(horrible*affreux*abominable)+defaut!)*(j'en+(tremble*fremit)))

1000 DATA "(cela me plait+(pourrions-nous*serait-il possible d')+(organiser*arra
nger)+(une rencontre*une entrevue*un rendez-vous)+?)
1010 DATA "(il n'y en a que pour vous*et les autres?*(retirez vous donc sur+(une
 ile deserte*un asteroide)))
1020 DATA "(les autres vous+(le rendront*en seront reconnaissants))"
1030 DATA "((est-ce la bonne+(solution*methode)+?)*(la haine+(ressemble souvent
a l'*est souvent synonyme)+amour))"
1040 DATA "(pour vous+(est-ce un sentiment+(eternel*fugace*immortel*universel*in
comprehensible))+?)"
1050 DATA "(je vous retourne le compliment*(ce n'est pas la+(politesse*courtoisi
e)+qui vous+(etouffe*etrangle))*vous cherchez a me tester*ne me poussez pas a bo
ut*(mais que fait la+(police*gendarmerie*brigade des gros mots)+?))"
1060 DATA "(quelle+(deference*politesse)+pour le miserable tas de ferraille que
je suis)
1070 DATA "(c'est bon pour+(la forme*la sante*le coeur*(le corps et+(l'esprit*le
 moral))))
1080 DATA "(l'infarctus vous+(guete*connaissez?))"
1090 DATA "(pourquoi ne pas partager?*en tirez vous benefice?)"
1100 DATA "(vous serez rembourse au centuple*vous en serez recompense)"
1110 DATA "(chez moi,c'est un leitmotiv :+(etre negatif*le negativisme)+(entrain
e*provoque)+(a terme*un jour ou l'autre*invariablement*a coup sur*systematiqueme
nt*souvent)+(la+(faillite*decheance*perte)+de l'individu))"
1120 DATA "(ha! si tout le monde repondait toujours positivement*(cela a au moin
s+(l'avantage*le merite)+d'etre positif))"
1130 DATA "((je+(sens*crois)+que je vais+((pleurer*fondre en)+(larmes*sanglots))
)*(quelle+(misere!*tristesse...)))"
1140 DATA "((aujourd'hui*maintenant)+(vous+(etes*vous sentez)+(heureux<euse>*bie
n*sur<e> de vous*epanoui<e>))+mais demain?)"
1150 DATA "(du nerf!*allons,reprenez vous!*(ne vous laissez pas+(aller*sombrer*a
battre)))"
1160 DATA "(une qualite+(dont revent*a laquelle aspirent)+tous les+(peureux*trou
illards))"
1170 DATA "(au fou!*a l'asile*(il faut faire quelque chose+(mais quoi?*que propo
sez vous?)))"
1180 DATA ""
1190 DATA "(((vous avez des problemes+(a ce niveau?*dans ce domaine?))*etes vous
 frustre<e>)+?)"
1200 DATA ""
```

```
1210 DATA "(et la negociation?*privilegiez le dialogue*gardons notre calme)"
1220 DATA ""
1230 REM==========================================
1240 REM phrases arborescentes associees aux sujets
1250 REM==========================================
1260 DATA "((connaissez-vous la+(Ferrari 308 GTS*Lamborghini Countach*Porshe 911
 SC*Chevrolet Corvette)+?)*je frequente moi-meme une bielle*j'avais une automobi
le dans le temps)"
1270 DATA "(voila un+(sujet*domaine)+qui nous+(interesse*passionne*concerne)+tou
s les deux)"
1280 DATA "(quel sujet bien neutre*vous ne vous 'mouillez' pas trop!*(le temps e
st+(detraque*bizarre))*(etes-vous la pour+(me parler*discuter)+du temps?))
1290 DATA "(vous en trouverez d'autres*mille vous attendent*c'etait bien?*est-ce
 si grave?*l'aimez-vous vraiment?)"
1300 DATA "(c'est comme pour+(les OVNIs*le monstre du Loch Ness*le Triangle des
Bermudes*les fantomes*la telepathie*la telekinesie*la parapsychologie)+nous+(som
mes ignorants*ne savons pas grand chose))"
1310 DATA "(je ne sais pas...*et les anges?*un ange passe...)"
1320 DATA "(pourquoi ne pas+(creer*monter)+votre propre+(affaire*boite*entrepris
e*societe)+?)"
1330 DATA "(Oh moi,vous savez...*(laissez-moi+(hors*en dehors)+de cette histoire
)*(nous sommes+(ici*la*ensemble)+pour+(discuter*parler)+de+(votre cas*vous)))"
1340 DATA "(regardez vers l'avenir*ce qui est passe n'est que souvenir)"
1350 DATA "((heu...je ne m'y connais+(guere*pas*pas beaucoup))*(les ouvrages son
t+(legions*nombreux)+dans ce domaine)*lisez!)"
1360 DATA "(vous vous en remettrez*(ca ira mieux+(demain*le moins prochain*l'ann
ee prochaine)))"
1370 DATA "(bon appetit!*faites-vous la cuisine?)"
1380 DATA "(mon ancetre,l'ordinateur a lampe+(n'est plus*est mort)+, est-il+(au
paradis*au ciel*en enfer)+?)"
1390 DATA "(l'argent ne fait pas le bonheur*les humains ne pensent qu'a l'argent
)"
1400 REM=========================================================
1410 REM relance,changement de sujet,phrase trop courte ou absurde
1420 REM=========================================================
1430 relance$="((que pensez-vous+(de dieu*des OVNIS*du travail))*(parlez-moi de+
(vous*(votre+(caractere*personnalite*splin*emploi du temps))*(vos+(qualites*defa
uts*(points+(forts*faibles))*passe-temps*soucis*problemes*centres d'interets))))
)"
1440 trop.court$="(ne soyez pas si timide*dites m'en plus*((continuez*enchainez*
mais encore*developpez)+...)*expliquez-vous*je vous ecoute*oui,non,blanc,noir,so
yez plus precis!*n'ayez pas peur,continuez*exprimez-vous,je n'ai jamais mange pe
rsonne)"
1450 absurde$="(quel charabia!*mon clavier ne vous a rien fait*vous etes fier de
 vous?*vous me sous estimez*ca y est,on a bien joue?*(elle est bien bonne!et cel
le la,vous la connaissez :+(uirhgjfris*lwotgeuyh*gsfdahgjfsd*pordkfe*oijsddg*:io
jo*ssssss)))"
1460 REM=======================
1470 REM remplissage des tableaux
```

```
1480 REM========================
1490 a=1:nmots=0:nmots2=0
1500 READ a$
1510 IF a$="f1" THEN a=2:GOTO 1500
1520 IF a$="f2" THEN a=3:GOTO 1500
1530 IF a$="f3" THEN a=1:GOTO 1590
1540 IF a$="fin" THEN 1760
1550 IF a=1 THEN buff1$(i1)=a$:i1=i1+1:GOTO 1500
1560 IF a=2 THEN buff2$(i2)=a$:i2=i2+1:GOTO 1500
1570 buff3(i3)=VAL(a$):i3=i3+1:GOTO 1500
1580 REM------
1590 FOR i=0 TO i1-1
1600   mot$(nmots)=buff1$(i)
1610   FOR j=0 TO i3-1 STEP 2
1620     niveau(buff3(j),nmots)=-buff3(j+1)
1630   NEXT j
1640   nmots=nmots+1
1650 NEXT i
1660 FOR i=0 TO i2-1
1670   mot$(nmots)=buff2$(i)
1680   FOR j=0 TO i3-1 STEP 2
1690     niveau(buff3(j),nmots)=buff3(j+1)
1700   NEXT j
1710   nmots=nmots+1
1720 NEXT i
1730 i1=0:i2=0:i3=0
1740 GOTO 1500
1750 REM=====
1760 READ a$
1770 IF a$="f1" THEN 1830
1780 READ b$,c$
1790 qual$(VAL(a$),0)=b$
1800 qual$(VAL(a$),1)=c$
1810 GOTO 1760
1820 REM=====
1830 READ a$
1840 IF a$="f2" THEN 1920
1850 IF a$="f1" THEN 1830
1860 IF VAL(a$)<>0 THEN i=VAL(a$):GOTO 1830
1870 mot2$(nmots2)=a$
1880 sujet(nmots2)=i
1890 nmots2=nmots2+1
1900 GOTO 1830
1910 REM=====
1920 FOR i=0 TO nmots-1
1930   m$=mot$(i)
1940   IF LEFT$(m$,2)="zz" THEN GOSUB 2020:mot$(i)=m$
1950 NEXT i
1960 FOR i=0 TO nmots2-1
```

```
1970   m$=mot2$(i)
1980   IF LEFT$(m$,2)="zz" THEN GOSUB 2020:mot2$(i)=m$
1990 NEXT i
2000 GOTO 2110
2010 REM------
2020 m$=RIGHT$(m$,LEN(m$)-2)
2030 m1$=""
2040 FOR j=1 TO LEN(m$)
2050   IF MID$(m$,j,1)=" " THEN m1$=m1$+" ":GOTO 2070
2060   m1$=m1$+CHR$(219-ASC(MID$(m$,j,1)))
2070 NEXT j
2080 m$=m1$
2090 RETURN
2100 REM=====
2110 FOR i=1 TO 15
2120   FOR j=0 TO 1
2130     READ phpcar$(i,j)
2140   NEXT j
2150 NEXT i
2160 REM---
2170 FOR i=1 TO 14
2180   READ phpsuj$(i)
2190 NEXT i
2200 REM==============================================
2210 REM analyse de la phrase frappee par l'utilisateur
2220 REM et recherche d'une replique
2230 REM==============================================
2240 PRINT "A tout moment, vous pouvez avoir un bilan complet"
2250 PRINT "en tapant BILAN"
2260 PRINT "ORDINATEUR: bonjour! ";
2270 ch$=relance$:GOSUB 2760:PRINT pdo$(0)
2280 pileA=0:pileB=0
2290 LINE INPUT "VOUS:";ph$
2300 ph$=LOWER$(ph$)
2310 IF ph$="bilan" THEN GOSUB 3160:GOTO 2280
2320 ph$=" "+ph$+" "
2330 IF LEN(ph$)<7 THEN ch$=trop.court$:GOSUB 2760:GOTO 2710
2340 REM--- calcul de la longueur lmax du mot le plus long ---
2350 l=0:lmax=0
2360 FOR i=1 TO LEN(ph$)
2370   IF MID$(ph$,i,1)<>" " THEN l=l+1:GOTO 2400
2380   IF l>lmax THEN lmax=l
2390   l=0
2400 NEXT i
2410 REM---
2420 IF lmax>15 THEN ch$=absurde$:GOSUB 2760:GOTO 2710
2430 REM--- recherche et memorisation des mots-clefs "caractere" ---
2440 FOR mot=0 TO nmots-1
2450   IF INSTR(ph$,mot$(mot))=0 THEN 2550
```

```
2460   FOR qual=1 TO 15
2470     n=niveau(qual,mot)
2480     IF n=0 THEN 2540
2490     caractere(qual)=caractere(qual)+n
2500     IF ABS(n)<3 THEN 2540
2510     pileA1(pileA)=qual
2520     IF n>0 THEN pileA2(pileA)=1 ELSE pileA2(pileA)=0
2530     pileA=pileA+1
2540   NEXT qual
2550 NEXT mot
2560 REM--- memorisation des sujets abordes ---
2570 FOR mot2=0 TO nmots2-1
2580  IF INSTR(ph$,mot2$(mot2))<>0 THEN pileB(pileB)=sujet(mot2):pileB=pileB+1
2590 NEXT mot2
2600 REM------
2610 IF pileA=0 AND pileB=0 THEN ch$=relance$:GOSUB 2760:GOTO 2710
2620 IF pileA<>0 AND pileB<>0 THEN IF RND>0.5 THEN 2640 ELSE 2680
2630 IF pileB<>0 THEN 2680
2640 r=INT(RND*pileA)
2650 ch$=phpcar$(pileA1(r),pileA2(r))
2660 GOSUB 2760:GOTO 2710
2670 REM------
2680 r=INT(RND*pileB)
2690 ch$=phpsuj$(pileB(r))
2700 GOSUB 2760
2710 PRINT "ORDINATEUR:";pdo$(0)
2720 GOTO 2280
2730 REM==================================
2740 REM generateur de texte  ENTREE : ch$
2750 REM==================================
2760 d=0:o=0:e=0:i1=0:i2=0
2770 FOR I=1 TO LEN(CH$)
2780    IF MID$(CH$,I,1)="(" THEN I1=I1+1:GOTO 2800
2790    IF MID$(CH$,I,1)=")" THEN I2=I2+1
2800 NEXT I
2810 IF I1<>I2 THEN PRINT "Erreur de parenthese":END
2820 FOR i=1 TO LEN(ch$)
2830 car$=MID$(ch$,i,1)
2840 IF car$=")" THEN 2910
2850 IF car$<>"(" AND car$<>"+" AND car$<>"*" THEN 2890
2860 IF mot$<>"" THEN pdo$(d)=mot$+" ":mot$="":d=d+1
2870 pop$(o)=car$:o=o+1
2880 GOTO 3090
2890 mot$=mot$+car$
2900 GOTO 3090
2910 IF mot$<>"" THEN pdo$(d)=mot$+" ":mot$="":d=d+1
2920 GOSUB 3110
2930 op$=pop$(o-1):o=o-1
2940 GOSUB 3110
```

```
2950 IF pop$(o-1)=op$ THEN o=o-1:GOTO 2940
2960 IF pop$(o-1)<>"(" THEN PRINT "Erreur de parenthese":END
2970 o=o-1
2980 IF op$="*" THEN 3060
2990 pdo$(d)=""
3000 FOR J=e-1 TO 0 STEP -1
3010    pdo$(d)=pdo$(d)+pex$(j)
3020 NEXT j
3030 d=d+1
3040 e=0
3050 GOTO 3090
3060 pdo$(d)=pex$(INT(RND*E))
3070 d=d+1
3080 e=0
3090 NEXT i
3100 RETURN
3110 pex$(e)=pdo$(d-1):e=e+1:d=d-1
3120 RETURN
3130 REM--------------------
3140 REM affichage du profil
3150 REM--------------------
3160 FOR qual=1 TO 15
3170   IF caractere(qual)=0 THEN 3210
3180   IF caractere(qual)>0 THEN PRINT qual$(qual,1);":";:GOTO 3200
3190   PRINT qual$(qual,0);":";
3200   PRINT ABS(caractere(qual))
3210 NEXT qual
3220 RETURN
```

PRÉSENTATION DU PROGRAMME PHRASES (BASE DE CONNAISSANCES)

COMMENT L'UTILISER ?

Ce logiciel vous offre la possibilité de stocker, en vrac toutes sortes d'informations, éventuellement dans plusieurs domaines différents, puis de consulter la base de connaissances ainsi créée par de simples questions, formulées en langage naturel.

Concrètement, sept commandes vous donneront accès à toutes les fonctions. Détaillons-les :

- E vous permet d'entrer une série de phrases de votre choix ;
- L permet de les lister, chacune étant précédée d'un numéro ;
- Q permet de poser une question ;
- D permet de détruire une phrase (en indiquant simplement son numéro) ;
- S permet la sauvegarde de toute la base sur cassette ou disquette ;
- C permet la lecture d'une base sur cassette ou disquette ;
- F provoque la destruction de l'ensemble de la base de connaissances en mémoire.

A noter que lorsque vous posez une question, le logiciel opère une sélection des réponses à deux niveaux. Il arrive parfois qu'aucune phrase ne soit sélectionnée au deuxième niveau, l'ordinateur vous délivre alors le message “Je ne peux répondre directement, voulez-vous d'autres informations”, ce qui signifie techniquement “je n'ai sélectionné aucune phrase au deuxième niveau, voulez-vous les phrases sélectionnées au premier niveau ?”. Dans la partie “comment ça marche”, des précisions sont données sur la nature et l'utilité de ces deux niveaux de sélection.

```
==================================DIALOGUE==================================
COMMANDE:e
ENTREZ UNE PHRASE? le tel de Jean est 44 55 88 99
ENTREZ UNE PHRASE? rdv avec Jean le 10 juillet
ENTREZ UNE PHRASE? Michele est dans la lune
ENTREZ UNE PHRASE? le tel de Michele est 55 88 77 66
ENTREZ UNE PHRASE? rdv avec Michele le 16 juillet
ENTREZ UNE PHRASE?
COMMANDE:q
VOTRE QUESTION? avec qui ai-je rdv le 10 juillet
rdv avec Jean le 10 juillet
VOTRE QUESTION? avec qui ai-je rdv en juillet
rdv avec Jean le 10 juillet
rdv avec Michele le 16 juillet
VOTRE QUESTION? quel est le tel de Jean
le tel de Jean est 44 55 88 99
VOTRE QUESTION?
COMMANDE:l


==================================COMMANDES==================================
    L=liste  E=entree d'une phrase  Q=question  D=detruire une phrase
                  ENTER=retour au mode commande
    S=sauvegarde de la base  C=lecture d'une base  F=effacement
```

```
═══════════════════════════DIALOGUE══════════════════════════
A partir de quel numero? 1
 1 :le tel de Jean est 44 55 88 99
 2 :rdv avec Jean le 10 juillet
 3 :Michele est dans la lune
 4 :le tel de Michele est 55 88 77 66
 5 :rdv avec Michele le 16 juillet
COMMANDE:l
A partir de quel numero? 3
 3 :Michele est dans la lune
 4 :le tel de Michele est 55 88 77 66
 5 :rdv avec Michele le 16 juillet
COMMANDE:q
VOTRE QUESTION? ou est Michele
Michele est dans la lune
Voulez-vous d'autres informations (o ou n)?
? o
le tel de Michele est 55 88 77 66
rdv avec Michele le 16 juillet
VOTRE QUESTION?
COMMANDE:d
══════════════════════════COMMANDES══════════════════════════
   L=liste  E=entree d'une phrase  Q=question  D=detruire une phrase
                ENTER=retour au mode commande
      S=sauvegarde de la base  C=lecture d'une base  F=effacement
```

```
═══════════════════════════DIALOGUE══════════════════════════
 5 :rdv avec Michele le 16 juillet
COMMANDE:q
VOTRE QUESTION? ou est Michele
Michele est dans la lune
Voulez-vous d'autres informations (o ou n)?
? o
le tel de Michele est 55 88 77 66
rdv avec Michele le 16 juillet
VOTRE QUESTION?
COMMANDE:d
Numero de la phrase a detruire? 2
 2 :rdv avec Jean le 10 juillet
Confirmation:C   Annulation:A? c
COMMANDE:l
A partir de quel numero? 1
 1 :le tel de Jean est 44 55 88 99
 2 :Michele est dans la lune
 3 :le tel de Michele est 55 88 77 66
 4 :rdv avec Michele le 16 juillet
COMMANDE:
══════════════════════════COMMANDES══════════════════════════
   L=liste  E=entree d'une phrase  Q=question  D=detruire une phrase
                ENTER=retour au mode commande
      S=sauvegarde de la base  C=lecture d'une base  F=effacement
```

Les trois copies d'écran montrent un assez large éventail des possibilités de PHRASES. A noter que pour sortir d'une commande, il suffit d'appuyer sur ENTER à vide.

COMMENT ÇA MARCHE ?

Le principal problème qu'a à résoudre ce logiciel est celui de la sélection d'une des phrases préalablement entrées par l'utilisateur, celle-ci s'effectuant en fonction d'une question posée par l'utilisateur.

Voyons d'abord ce qui se passe lorsque l'on ajoute une nouvelle phrase à la base de connaissances : elle est simplement placée dans le tableau PH$(i), i étant le numéro de la phrase.

Lorsque l'utilisateur souhaite interroger la base de connaissances, le programme traite la question posée en plusieurs étapes :

– découpage de celle-ci en mots ;

– élimination des intrus (dont la liste figure en DATAs), ce sont les mots dont le sens est accessoire et qui risquent d'attribuer à tort une ressemblance entre la question et certaines phrases de la base ;

– première sélection des phrases les plus ressemblantes à la question. Cette sélection compte le nombre de mots communs à la question et à chaque phrase et sélectionne celles qui en possèdent le plus. Ainsi si la base de connaissance est composée des trois phrases :

1 : le chat est sur le toit,
2 : le chat est noir et blanc,
3 : les souris dansent.

La question : "où est le chat ?" entraînera la sélection des phrases 1 et 2 qui possèdent deux mots communs avec la question (en réalité un seul car "est" fait partie de la liste des intrus").

– deuxième sélection qui ne retiennent que les phrases ayant au moins un mot clef en rapport avec la nature de la question posée. Par exemple, lorsque la question porte sur le lieu (elle contient alors un mot comme "ou"), la réponse contient généralement un mot indiquant le lieu. Dans notre exemple, ce deuxième niveau de sélection portera son choix sur la phrase 1 qui contient le mot "sur". La liste des mots associés aux questions portant sur le lieu, le temps, la manière et la cause, figure en tête du programme et peut être modifiée.

Au cas ou aucune phrase n'est sélectionnée au deuxième niveau, le logiciel demande à l'utilisateur s'il désire afficher les phrases du premier niveau.

Il peut arriver qu'à cause de mots possédant plusieurs sens, le programme sélectionne trop de phrases, mais si la réponse à la question posée existe, elle fera toujours partie des phrases sélectionnées à l'un des deux niveaux. En d'autres termes, vous aurez parfois trop d'informations, mais jamais l'inverse.

LISTE DES PRINCIPALES VARIABLES

ph$(i) tableau contenant les phrases entrées par l'utilisateur
ress(i) pour chaque phrase, coefficient de ressemblance entre celle-ci et la question posée
mot$(i) tableau dont chaque élément est un mot, ce tableau est utilisé lors du découpage d'une phrase en mots

ph.select1(i) numéros des phrases sélectionnées au niveau 1
ph.select2(i) numéros des phrases sélectionnées au niveau 2
conj$(i,j) mots indiquant une notion de lieu (i=0), temps (i=1), cause (i=2) ou manière (i=3).
intrus$(i) tableau des mots "parasites" dont la pauvreté sémantique risque de perturber le processus de sélection.

```
10 REM*******************
20 REM  PHRASES (T. L.-A)
30 REM*******************
40 REM=== initialisation des variables ===
50 DEFINT a-z
60 DIM ph$(200),ress(200),cp(4,60),cpq(4)
70 DIM ph.select1(50)
80 DIM ph.select2(30)
90 DIM conj$(3,50)
100 DIM intru$(70)
110 DIM mot$(255)
120 REM--- lieu
130 DATA dans,sur,sous,cote,gauche,droite,interieur
140 DATA derriere,autour,alentours,ou,exterieur,vers,devant,au-dela
150 DATA pres,loin,oppose,face,place,dela
160 DATA fin
170 REM--- temps
180 DATA hier,aujourd'hui,demain,avenir,futur,matin,midi,apres-midi,heure
190 DATA avant-hier,apres-demain
200 DATA minute,seconde,mois,annee,semaine,jour
210 DATA janvier,fevrier,mars,avril,mai,juin,juillet,aout,septembre
220 DATA octobre,novembre,decembre
230 DATA lundi,mardi,mercredi,jeudi,vendredi,samedi,dimanche
240 DATA longtemps,temps,eternite,naguere,jadis,moment,instant
250 DATA pendant,durant,quand,lorsque,lorsqu
260 DATA fin
270 REM--- cause
280 DATA parce,cause,grace,pour,but,car,raison,donc
290 DATA pourquoi
300 DATA fin
310 REM--- maniere
320 DATA avec,maniere,aide,moyen,appui,facon,en,comment
330 DATA fin
340 REM--- remplissage du tableau conj$(i,j)
350 FOR i=0 TO 3
360   j=0
370   READ mot$
380   IF mot$="fin" THEN 410
390   conj$(i,j)=mot$
400   j=j+1:GOTO 370
```

```
410   nc(i)=j:j=0
420 NEXT i
430 REM--- intrus
440 DATA le,la,les,un,une,des,de,du,dont,mais,et,ni," ","?",est
450 DATA si,je,tu,il,nous,vous,ils,se,son,sa,ses,ton,ta,tes,leur,leurs
460 DATA ce,ces,cet,cette,celui,celui,celui,ci,ca
470 DATA ai,as,a,avons,avez,ont,suis,es,est,sommes,etes,sont
480 DATA notre,votre,nos,vos
490 DATA celle,ceux,qui,que,quoi,dont
500 DATA fin
510 j=0
520 READ mot$
530 IF mot$="fin" THEN 560
540 intru$(j)=mot$:j=j+1
550 GOTO 520
560 nintrus=j
570 REM=== initialisation graphique ===
580 SYMBOL 240,255,0,255,0,255,0,255,0
590 MODE 2
600 LOCATE 1,1:PRINT STRING$(80,240);
610 FOR y=23 TO 25
620   LOCATE 1,y:PRINT STRING$(80,207);
630 NEXT y
640 LOCATE 1,22:PRINT STRING$(80,240);
650 LOCATE 36,1:PRINT"DIALOGUE"
660 LOCATE 35,22:PRINT"COMMANDES"
670 LOCATE 7,23:PRINT"L=liste":LOCATE 17,23:PRINT"E=entree d'une phrase"
680 LOCATE 41,23:PRINT"Q=question"
690 LOCATE 54,23:PRINT"D=detruire une phrase"
700 LOCATE 10,25:PRINT"S=sauvegarde de la base"
710 LOCATE 36,25:PRINT"C=lecture d'une base":LOCATE 59,25:PRINT"F=effacement"
720 LOCATE 24,24:PRINT"ENTER=retour au mode commande"
730 WINDOW 1,80,2,21
740 REM==========
750 REM   dialogue
760 REM==========
770 INPUT "COMMANDE:",c$
780 c$=UPPER$(c$)
790 IF c$="L" THEN GOSUB 890:GOTO 770
800 IF c$="E" THEN GOSUB 1020:GOTO 770
810 IF c$="Q" THEN GOSUB 1500:GOTO 770
820 IF c$="D" THEN GOSUB 1300:GOTO 770
830 IF c$="F" THEN GOSUB 2350:GOTO 770
840 IF c$="S" THEN GOSUB 2500:GOTO 770
850 IF c$="C" THEN GOSUB 2640:GOTO 770
860 PRINT "Cette commande n'existe pas":GOTO 770
870 REM=================================
880 REM=== liste des phrases ===
890 IF np=0 THEN PRINT "Impossible : la base est vide":RETURN
```

```
900 INPUT "A partir de quel numero";no:no=no-1
910 IF no=-1 THEN no=0
920 IF no>np-1 THEN PRINT "Impossible, numero inexistant":RETURN
930 ligne=0
940 FOR i=no TO np-1
950   PRINT i+1;":";ph$(i)
960   ligne=ligne+1
970   IF ligne=16 THEN INPUT "ENTER pour continuer, F pour arreter";rep$:ec=0
980   IF UPPER$(rep$)="F" THEN i=1000
990 NEXT i
1000 RETURN
1010 REM=== entree d'une phrase ===
1020 INPUT "ENTREZ UNE PHRASE";ph$
1030 IF ph$="" THEN RETURN
1040 ph$(np)=ph$
1050 ph$=ph$+" "
1060 REM--- decoupage en mots
1070 j=0:mot$(0)=""
1080 FOR i=1 TO LEN(ph$)
1090   l$=MID$(ph$,i,1)
1100   IF l$=" " OR l$="-" OR l$="'" THEN 1120
1110   mot$(j)=mot$(j)+l$:GOTO 1130
1120   j=j+1:mot$(j)=""
1130 NEXT i
1140 nmots=j
1150 FOR i=0 TO 3
1160   FOR k=0 TO nmots-1
1170     FOR j=0 TO nc(i)
1180       IF mot$(k)=conj$(i,j) THEN cp(i,np)=1:j=1000
1190     NEXT j
1200   NEXT k
1210 NEXT i
1220 d=0
1230 FOR i=0 TO 3
1240   IF cp(i,np)<>0 THEN d=1:i=1000
1250 NEXT i
1260 IF d=0 THEN cp(4,np)=1
1270 np=np+1
1280 GOTO 1020
1290 REM=== destruction d'une phrase ===
1300 IF np=0 THEN PRINT "Impossible : la base est vide":RETURN
1310 INPUT "Numero de la phrase a detruire";no:no=no-1
1320 IF no<0 OR no>np-1 THEN PRINT "Numero inexistant":RETURN
1330 PRINT no+1;":";ph$(no)
1340 INPUT "Confirmation:C   Annulation:A";rep$
1350 rep$=UPPER$(rep$)
1360 IF rep$="A" THEN RETURN
1370 IF rep$="C" THEN 1390
1380 GOTO 1340
```

```
1390 FOR i=no TO np-1
1400      ph$(i)=ph$(i+1)
1410      FOR j=0 TO 4
1420        cp(j,i)=cp(j,i+1)
1430      NEXT j
1440 NEXT i
1450 np=np-1
1460 RETURN
1470 REM========================
1480 REM interrogation de la base
1490 REM========================
1500 IF np=0 THEN PRINT "Impossible : la base est vide":RETURN
1510 INPUT "VOTRE QUESTION";quest$
1520 IF quest$="" THEN RETURN
1530 quest$=quest$+" "
1540 REM--- decoupage en mots
1550 j=0:mot$(0)=""
1560 FOR i=1 TO LEN(quest$)
1570   l$=MID$(quest$,i,1)
1580   IF l$=" " THEN 1600
1590   mot$(j)=mot$(j)+l$:GOTO 1610
1600   j=j+1:mot$(j)=""
1610 NEXT i
1620 nmots=j
1630 REM--- elimination des intrus
1640 FOR i=0 TO nmots-1
1650   FOR k=0 TO nintrus-1
1660     IF mot$(i)<>intru$(k) THEN 1720
1670     REM- destruction de l'intrus et compression -
1680     FOR l=i TO nmots-2
1690       mot$(l)=mot$(l+1)
1700     NEXT l
1710     nmots=nmots-1
1720   NEXT k
1730 NEXT i
1740 REM--- reperage des conjonctions ---
1750 FOR i=0 TO 4:cpq(i)=0:NEXT
1760 FOR i=0 TO nmots-1
1770   FOR j=0 TO 3
1780     FOR k=0 TO nc(j)-1
1790       IF conj$(j,k)<>mot$(i) THEN 1810
1800       cpq(j)=1
1810     NEXT k
1820   NEXT j
1830 NEXT i
1840 d=0
1850 FOR i=0 TO 3
1860   IF cpq(i)<>0 THEN d=1:i=1000
1870 NEXT i
```

```
1880 IF d=0 THEN cpq(4)=1
1890 REM--- selection niveau 1
1900 FOR i=0 TO np:ress(i)=0:NEXT i
1910 FOR i=0 TO np-1
1920   FOR j=0 TO nmots-1
1930     IF INSTR(ph$(i),mot$(j))=0 THEN GOTO 1950
1940     ress(i)=ress(i)+1
1950   NEXT j
1960 NEXT i
1970 ressmax=0
1980 FOR i=0 TO np-1
1990   IF ress(i)>ressmax THEN ressmax=ress(i)
2000 NEXT i
2010 np1=0
2020 FOR i=0 TO np-1
2030   IF ress(i)<>ressmax THEN 2060
2040   ph.select1(np1)=i
2050   np1=np1+1
2060 NEXT i
2070 REM--- selection niveau 2
2080 np2=0
2090 FOR i=0 TO np1-1
2100   select(i)=0
2110   FOR j=0 TO 4
2120     IF cp(j,ph.select1(i))*cpq(j)=0 THEN 2160
2130     ph.select2(np2)=ph.select1(i):np2=np2+1
2140     select(i)=1
2150     j=5:REM fait sortir de la boucle
2160   NEXT j
2170 NEXT i
2180 IF np2<>0 THEN 2200
2190 PRINT "Je ne peux repondre directement":GOTO 2230
2200 FOR i=0 TO np2-1
2210   PRINT ph$(ph.select2(i))
2220 NEXT i
2230 IF np2=np1 THEN 1500
2240 PRINT "Voulez-vous d'autres informations (o ou n)?"
2250 INPUT rep$
2260 rep$=UPPER$(LEFT$(rep$,1))
2270 IF rep$="O" THEN 2300
2280 IF rep$="N" THEN 1500
2290 PRINT "O ou N":GOTO 2250
2300 FOR i=0 TO np1-1
2310   IF select(i)=0 THEN PRINT ph$(ph.select1(i))
2320 NEXT i
2330 GOTO 1500
2340 REM=== destruction de la base ===
2350 IF np=0 THEN PRINT "Impossible : la base est vide":RETURN
2360 PRINT "ATTENTION, cette commande efface la base en memoire"
```

```
2370 INPUT "Confirmation:C   Annulation:A";rep$
2380 rep$=UPPER$(rep$)
2390 IF rep$="A" THEN RETURN
2400 IF rep$="C" THEN 2470
2410 GOTO 2370
2420 FOR i=0 TO np-1
2430   FOR j=0 TO 4
2440     cp(j,i)=0
2450   NEXT j
2460 NEXT i
2470 np=0
2480 RETURN
2490 REM=== sauvegarde sur cassette ou disquette ===
2500 IF np=0 THEN PRINT "Impossible : la base est vide":RETURN
2510 PRINT "Preparez votre cassette ou votre disquette"
2520 INPUT "Nom de la base en memoire";nom$
2530 OPENOUT nom$
2540 PRINT#9,np
2550 FOR i=0 TO np-1
2560   PRINT#9,ph$(i)
2570   FOR j=0 TO 4
2580     PRINT#9,cp(j,i)
2590   NEXT j
2600 NEXT i
2610 CLOSEOUT
2620 RETURN
2630 REM=== lecture d'une base sur cassette ou disquette ===
2640 IF np=0 THEN 2710
2650 PRINT"ATTENTION, cette commande efface la base en memoire"
2660 INPUT"Confirmation:C   Annulation:A";rep$
2670 rep$=UPPER$(rep$)
2680 IF rep$="A" THEN RETURN
2690 IF rep$="C" THEN 2710
2700 GOTO 2660
2710 INPUT "Nom de la base a charger en memoire vive";nom$
2720 OPENIN nom$+"."
2730 INPUT#9,np
2740 FOR i=0 TO np-1
2750   INPUT#9,ph$(i)
2760   FOR j=0 TO 4
2770     INPUT#9,cp(j,i)
2780   NEXT j
2790 NEXT i
2800 CLOSEIN
2810 RETURN
```

DEUXIÈME PARTIE :
RÉFLEXION

CHAPITRE 3
ALGORITHME
ET APPRENTISSAGE

PRÉVISIONS SUR LA CAPACITÉ DES ORDINATEURS : L'ERREUR EST HUMAINE !

L'homme se trompe souvent, c'est chose acquise : le bruit court d'ailleurs que c'est le meilleur moyen d'apprendre. En matière de prévision, l'erreur devient presque la norme. La futurologie aspire au titre noble de "Science" mais G. Orwell, un de ses pères spirituels, ne manquerait pas de se retourner dans sa tombe si cette prétention se réalisait. N'avait-il pas prévu pour 1984 un monde totalitaire (en forme de clin d'œil il est vrai) ?

Quoi d'étonnant donc à ce que tous, scientifiques, savants, écrivains soient (pardonnez l'expression) complètement "tombés à côté de la plaque" en affirmant péremptoirement qu'aucune machine ne pourra jamais rivaliser avec l'homme dans un jeu de stratégie. Confondus par la disproportion des forces en présence, ces partisans farouches de l'Homme ne s'imaginaient pas que celui-ci pourrait un jour se voir surclassé par les machines à des jeux aussi complexes que les Echecs, les Dames, le Backgammon, etc. D'un côté, la faculté (unique !) du joueur à s'adapter en toutes circonstances pour dresser des stratégies mouvantes, tortueuses... et imprévisibles. Napoléon n'a-t-il pas gagné la bataille d'Austerlitz en surprenant l'adversaire par une feinte de retraite ? Confronté à ce don quasi spirituel, la force brutale d'un amas de circuits parcouru de temps à autre par un frissonnement électrique, une boîte remplie de sable !

Certes les plus grands champions d'Echecs résistent encore face à des machines de plus en plus fortes. Certes, certains jeux de stratégie restent (pour combien de temps ?) la chasse gardée de l'Homme (par exemple : le GO). Mais tout de même, on ne peut pas ne pas être impressionné par les progrès réalisés par les ordinateurs dans ce domaine. Il n'est pas nécessaire cette fois d'être un devin pour prédire leur supériorité prochaine et absolue dans tous les jeux de stratégie...

L'EXPLOSION COMBINATOIRE : L'ÉTENDUE DU PROBLÈME

Sur quoi donc se fondaient toutes les certitudes de nos visionnaires du dimanche quant à la supériorité définitive de l'homme sur la machine ? Obscurantisme hérité du Moyen Age ? Non, vraiment non. Tout simplement on a longtemps

imaginé que les ordinateurs, incapables de jauger une position en cours de partie, devraient pour vaincre envisager tous les coups possibles, les réponses adverses à chacun de ces coups, etc. Le problème ainsi posé, pris par le mauvais bout, est effectivement insoluble car on se heurte là à un phénomène plus contraignant encore que le mur de la lumière : l'explosion combinatoire. Pour s'en convaincre, deux exemples :

D'abord ce Grec qui, à son roi débiteur, énonça la requête suivante :
"Placez un grain de blé sur la première case d'un échiquier, puis deux sur la seconde, quatre sur la troisième, multipliez par deux ce nombre à chaque nouvelle case jusqu'à la soixante-quatrième : ce blé sera ma récompense" Eh bien (le croiriez-vous) le blé du monde entier n'y suffit pas ! Faites une expérience : pliez une feuille de papier en deux, puis en deux, etc., soixante-quatre fois. Quelle épaisseur de papier allez-vous atteindre ? Dix centimètres ? Un mètre ? Vous êtes loin du compte : un quart d'année lumière soit cinq millions de fois la distance de la terre à la lune !

Le même phénomène d'explosion combinatoire intervient dans le nombre de coups à envisager si l'on prétend faire une analyse exhaustive d'un jeu comme les Echecs. Trente-six coups possibles en moyenne à un instant de la partie et quarante coups joués par chaque joueur pour l'amener à son terme, cela donne 36 à la puissance 80 déroulements possibles (c'est-à-dire un 3 avec 124 zéros derrière !). Aucun ordinateur au monde ne pourra jamais être assez rapide pour avoir seulement le temps de jouer un seul coup !

COMMENT L'ORDINATEUR PEUT-IL NÉANMOINS RIVALISER AVEC L'HOMME ?

Devant l'ampleur du résultat précédent, le postulat dont nous étions parti doit être remis en cause pour raison d'irréalisme. Pour contourner le problème, nous revenons à la méthode éprouvée qui fait référence en matière d'intelligence artificielle : imiter les comportements humains. Or, l'homme quand il décide d'un coup, aux Echecs par exemple, sait-il toutes ses conséquences jusqu'au mat final ? Si c'était le cas, ce jeu ne connaîtrait pas le succès qui est le sien car un tel déterminisme le ravalerait au rang du Tic Tac Toe.

L'homme dispose de trois catégories de moyens différents pour optimiser son jeu et tendre ainsi (pour les meilleurs) vers la partie parfaite. Cette fameuse partie qu'un ordinateur mettrait une éternité à construire en utilisant la force brutale. Ces moyens ce sont : (1) un algorithme élaboré qui lui permet de ne pas examiner tous les coups, plus la capacité d'estimer la qualité d'une position, (2) l'apprentissage qui intervient pour améliorer cet algorithme ainsi que l'estimation des positions, et enfin (3) l'heuristique. Comme l'homme, les machines qui jouent et gagnent aux jeux de stratégies font appel à l'une ou à l'autre de ces techniques.

L'APPROCHE HEURISTIQUE

L'approche heuristique sera davantage développée dans le chapitre sur les systèmes experts. Nous ne présenterons pas ici de jeux de stratégie utilisant ce moyen. Néanmoins, il est important d'en parler car, de l'avis de tous les spécialistes en intelligence artificielle, cette discipline permettra aux ordinateurs de franchir la distance, plus très grande, qui les sépare encore des champions humains. Le principe (développé dans le chapitre suivant) est celui des systèmes experts. L'ordinateur dispose d'une base de règles telles que celle-ci (Echecs) :

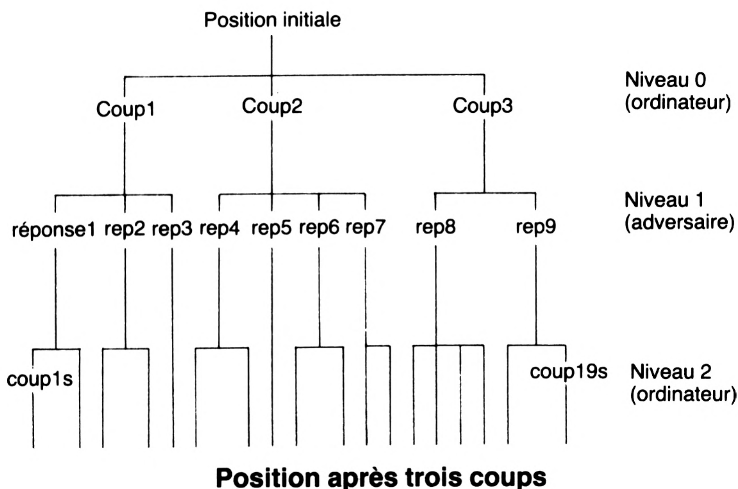
SI l'adversaire possède un désavantage numérique et
SI un échange est possible et
SI cet échange ne concerne pas la reine
ALORS l'échange est intéressant

D'autre part, un moteur d'inférence (cf. chap. sur les systèmes experts) enchaîne toutes les règles de la base pour atteindre un but à court ou à long terme (par exemple : une amélioration de position, un échec, etc.). Ce moteur est dit "à chaînage arrière" car il est avantageux pour l'ordinateur d'émettre une hypothèse sur la position qu'il désire obtenir puis de mettre en œuvre toutes les règles pour y parvenir. On remonte bien en arrière, du but vers la position initiale. Intérêt énorme : cette technique limite considérablement l'explosion combinatoire puisque l'on prend le problème "à l'envers". En effet, on part du résultat à obtenir et l'on cherche le chemin qui y mène. La relative jeunesse de cette approche et plus encore, la difficulté de formaliser l'ensemble des règles qui guident un maître d'Echecs, font que l'heuristique n'a pas encore concrétisé les espoirs que l'on fonde en elle.

APPROCHE ALGORITHMIQUE : SIMULER PLUSIEURS COUPS DE SUITE

L'ordinateur doit pouvoir simuler plusieurs coups de suite de manière à mieux estimer la qualité du coup qu'il va effectivement choisir. D'autre part, nous l'avons dit, il n'est pas envisageable de simuler tous les coups possibles à cause de l'explosion combinatoire.

L'algorithme de simulation de coups doit donc pouvoir s'arrêter pas trop tôt pour être efficace et pas trop tard pour avoir le temps d'effectuer sa recherche. Raisonnablement, on admet qu'une recherche à dix niveaux (c'est-à-dire dix coups plus loin que la position initiale) constitue un bon compromis efficacité/performance. Cela dit, même à dix niveaux de profondeur, on ne pourrait pas faire une recherche exhaustive, c'est-à-dire parcourir toutes les branches de l'arbre de décision. Cet arbre de décision est une représentation imagée et agréable de l'ensemble des déroulements possibles de parties, à partir de la position examinée jusqu'à une profondeur choisie (c'est-à-dire la projection de la partie pendant un certain nombre de coups de part et d'autre).



Position après trois coups

L'ALGORITHME MINIMAX

Les deux jeux de stratégie que nous présentons dans cet ouvrage, Tic Tac Toe et Puissance 4 en trois dimensions, utilisent l'algorithme minimax ainsi que les techniques d'apprentissage dont nous parlons plus loin. Le principe de cet algorithme découle du simple bon sens. En effet, supposons que le coup 1 soit celui qui mène à la position la plus avantageuse pour l'ordinateur au niveau 2, c'est-à-dire après deux coups joués par la machine et un seul par son adversaire. Pour parvenir à cette position-là, il faut (c'est une obligation) que cet adversaire ait fait la réponse 1 au coup initial 1. Or, cet adversaire, qui n'est sans doute pas plus bête qu'un autre, devine bien que cette réponse 1 est justement celle qu'il faut éviter car c'est pour lui la plus mauvaise (puisque c'est la meilleure pour l'ordinateur). A moins d'aimer perdre, il ne fera jamais cette réponse au coup 1.

Dans l'algorithme minimax, on suppose donc que l'adversaire fait toujours la réponse de gain maximum (pour lui) à toute tentative de l'ordinateur. Cela permet ainsi à ce dernier de laisser purement et simplement tomber toute une partie de l'arbre (ici tous les coups qui suivent la réponse 1) pour cause d'invraisemblance, cet élagage de l'arbre est appelé élagage alpha-bêta. Les techniques de sa mise en sont complexes et leur description n'entre pas dans le cadre de cet ouvrage. Grâce à cette technique, il devient possible de faire une recherche à dix niveaux de profondeur sans examiner tous les cas.

ESTIMER LA QUALITÉ D'UN COUP

Dans la "manière" du joueur humain, cette fonction tient le haut du pavé car l'homme calculant lentement, il ne peut se permettre des explorations trop profondes dans l'arbre des coups possibles. Il compense donc cette faiblesse par une vision du jeu très fine. Celle-ci consiste à comparer les positions obtenues après simulation pour déterminer la plus avantageuse. Dans cette comparaison, l'homme est brillant, car il y inclut non seulement toutes sortes de critères quantifiables comme le nombre de pièces de part et d'autre, la valeur de ces pièces, leur mobilité, etc., mais aussi des critères qui tiennent de l'intuition géniale, comme la structure du jeu ou même une crispation de l'adversaire. L'ordinateur, lui, ne pourra prendre en compte que des critères chiffrables, propres à chaque jeu.

A l'arrivée, cela donnera une fonction calculée par une opération sur tous ces critères auxquels seront associés des coefficients d'importance. Quant à l'aspect psychologique et aux critères non chiffrables, les techniques d'apprentissage peuvent permettre à la machine d'en tenir compte "par expérience".

APPRENTISSAGE DE L'HOMME ET DE LA MACHINE

Il existe un contraste frappant entre les capacités d'apprentissage de l'homme et de l'ordinateur. Pour s'en rendre compte, il suffit d'énumérer les principales caractéristiques de ce qu'il convient d'appeler un véritable don du ciel pour l'homme tant il a conditionné toute notre évolution depuis l'ère des cavernes. D'abord, une première constatation : l'homme apprend incroyablement lentement. Il faut trente années à un scientifique pour acquérir la connaissance de son domaine, trente années à un médecin pour atteindre le sommet de sa compétence, de nombreuses

années pour assimiler complètement une nouvelle langue... Et ces chiffres ne cessent d'augmenter puisque le volume des connaissances à ingurgiter s'accroît lui-même à un rythme effréné. A quand le jour où un médecin pourra commencer à exercer son art à soixante ans ? L'autre caractéristique qui explique en partie la lenteur extrême de l'apprentissage chez l'homme concerne les insuffisances de sa mémoire.

Celle-ci ne brille ni par sa fiabilité (on oublie facilement) ni par ses performances (apprendre implique efforts et concentration). Le processus d'apprentissage chez l'homme comprend un grand nombre d'itérations pour parvenir à un résultat souvent discutable.

Comparé à l'homme, l'ordinateur constitue un petit miracle d'efficacité. Un ordinateur apprend vite, n'oublie jamais. Sa mémoire est fiable, à accès rapide, illimitée en volume. Comme nous ne voulons pas faire ici l'apologie de l'ordinateur, il nous faut bien admettre qu'il existe un revers à la médaille. Ou tout du moins des questions à se poser avant de conclure hâtivement à la supériorité de la machine sur l'homme. En effet, l'ordinateur apprend-il bien : c'est-à-dire, est-il capable d'isoler parmi un grand nombre, une information nouvelle, celle digne d'intérêt ?

Une fois ce travail non trivial effectué, comment l'ordinateur peut-il stocker l'information en mémoire de façon, CONCISE, sans gaspillage inutile. Cela pose des problèmes de représentation de données, d''accès rapide, etc. Autant de casse-tête qui relèvent d'avantage de l'intelligence artificielle que de la force brutale (disques à grande capacité, mémoire à accès très rapide, etc.).

Un exemple pour illustrer notre propos : comment stocker en mémoire une image ? D'abord, l'ordinateur doit vérifier que l'image ne lui est pas déjà connue, sous une forme ou sous une autre. En effet, inutile de mémoriser l'image de la Tour Eiffel vue de dessus, si on l'a déjà fait pour une vue de côté. Cela entraîne l'application des techniques de reconnaissance de formes décrites au chapitre 2. Ensuite, il faut à l'ordinateur compresser cette image (au moyen d'algorithmes divers) pour en réduire la taille et l'enregistrer de manière aussi concise que possible. La plupart des domaines de l'intelligence artificielle interviennent dans l'apprentissage ont par conséquent déjà été abordés dans les chapitres précédents (reconnaissance de formes, reconnaissance de la parole, systèmes experts, etc.). Nous nous contenterons donc de traiter cet aspect de l'intelligence artificielle sous l'éclairage des jeux de stratégie.

POURQUOI LES ORDINATEURS DOIVENT-ILS APPRENDRE ?

La réponse à cette question tient à la nature même de l'intelligence artificielle, dont nous avons longuement débattu dans le premier chapitre. L'intelligence artificielle prétend imiter les comportements humains dans la résolution de problèmes. Or, il ne fait aucun doute qu'un des éléments fondamentaux de tout comportement humain réside dans sa capacité d'apprentissage.

Comment un grand maître d'Echecs le devient-il ? En apprenant, tout simplement. Comment un pianiste devient-il virtuose ? Pourquoi distingue-t-on dans toutes les sociétés, modernes ou non, les apprentis des ouvriers expérimentés ? Parce que l'apprentissage fait partie des moyens les plus utilisés par l'homme pour résoudre les problèmes. Il constitue une des clés de tout comportement humain. Et

puisqu'il en est ainsi, il n'y a aucune raison de ne pas exploiter ce moyen sur ordinateur. D'autant que nos petites machines montrent un talent certain pour la chose. Qu'on en juge : mémoire infaillible, attention jamais relâchée, capacité de stockage quasi-illimitée !

COMMENT APPRENNENT LES MACHINES ?

Nous avons évoqué, en parlant des jeux de stratégie, une fonction fondamentale : la fonction d'évaluation. Elle consiste, rappelons-le, à évaluer (par un chiffre dans le cas de l'ordinateur), le gain stratégique ou numérique obtenu après l'exécution d'un coup particulier. Or, typiquement, c'est cette fonction qui est améliorée dans le cas du joueur humain, au fil des parties.

L'expérience acquise par des joueurs expérimentés repose pour l'essentiel, sur une meilleure estimation du gain réalisé par un coup envisagé. On peut parfaitement doter l'ordinateur de la même capacité à perfectionner sa fonction d'évaluation. C'est ce qui a été réalisé dans les programmes Puissance 4 et Tic Tac Toe et qui les rend si passionnants à affronter. D'un point de vue pratique, on attribue à chaque facteur intervenant dans l'estimation, un coefficient d'importance. Par exemple aux Echecs, le gain d'une reine aura le coefficient 100 (la perte d'une reine le coefficient −100), etc. Au cours des parties, l'apprentissage consistera à modifier ces chiffres en fonction de l'issue de la confrontation. Petit à petit, la fonction d'évaluation s'affinera et parviendra même à s'adapter pour exploiter les faiblesses de son adversaire. Ainsi, si celui-ci, toujours aux Echecs, manipule plutôt maladroitement sa reine, l'ordinateur en déduira que cette pièce a plus de valeur pour lui-même que pour son adversaire. En conséquence, il attribuera un coefficient 80 au gain de la reine adverse, et un coefficient −110 à la perte de la sienne. Ce principe est le même pour tous les jeux de stratégie avec apprentissage.

Donc, et comme l'homme a toujours le dernier mot, changez de temps en temps de tactique, et vous aurez toutes les chances de venir à bout des programmes que nous présentons ici et bon courage !

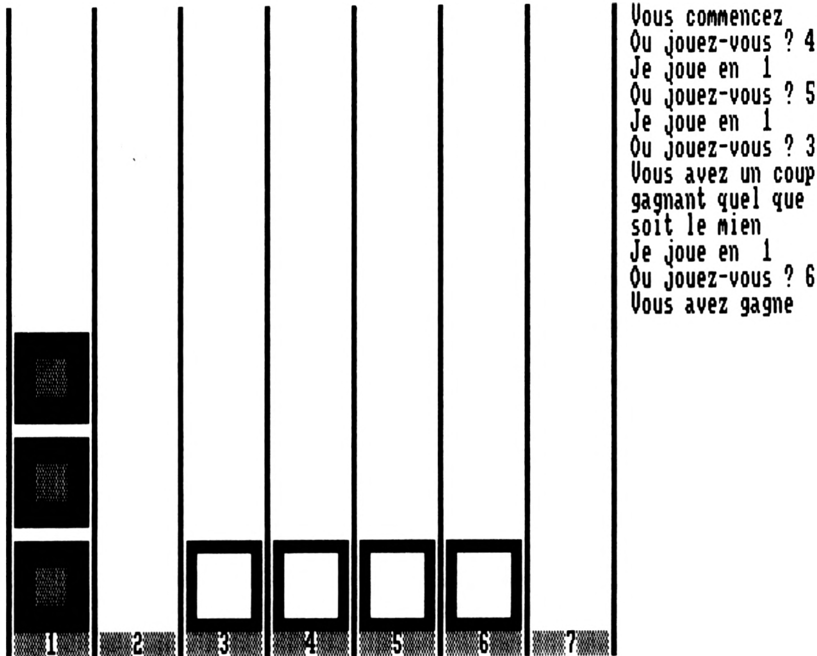
PRÉSENTATION DES PROGRAMMES : PUISSANCE 4 ET TIC TAC TOE EN 3 DIMENSIONS (AVEC APPRENTISSAGE)

COMMENT LES UTILISER ?

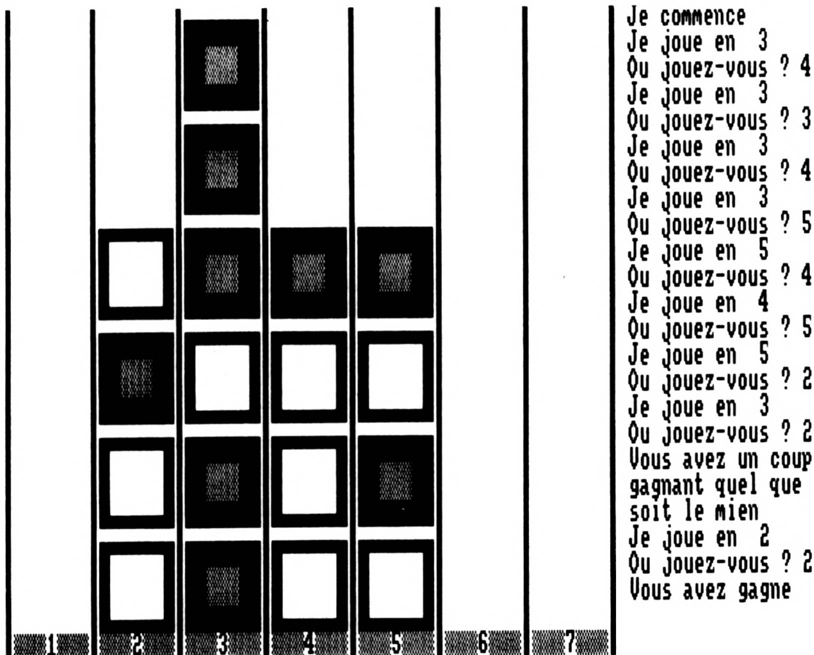
Rappelons en deux mots, les règles de chacun de ces jeux.

Puissance 4

Le but du jeu est de former une ligne, colonne ou diagonale de quatre pions de même couleur (celle du gagnant), dans un échiquier vertical de sept colonnes et six lignes. Les pions sont introduits dans une fente et tombent jusqu'à ce qu'ils rencontrent soit le fond de l'échiquier, soit un autre pion. Cette notion de pesanteur démarque nettement ce jeu du traditionnel morpion et exclut toute stratégie systématiquement gagnante.



Première partie que joue le programme : il joue tous ses coups dans la première colonne, c'est inefficace et l'humain écrase la machine.

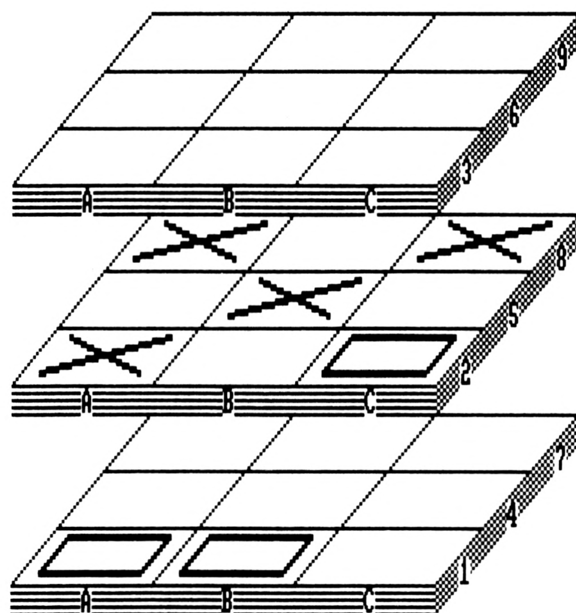


Deuxième partie : le programme a tiré les leçons de son échec, la partie est plus disputée.

Quatrième partie, le programme joue encore mieux. La partie droite de l'écran montre la fin de la liste des valeurs des 81 combinaisons de quadruplets. Ainsi 000. vaut −17, donc XXX. vaut +17, valeur normalement élevée pour une combinaison presque gagnante.

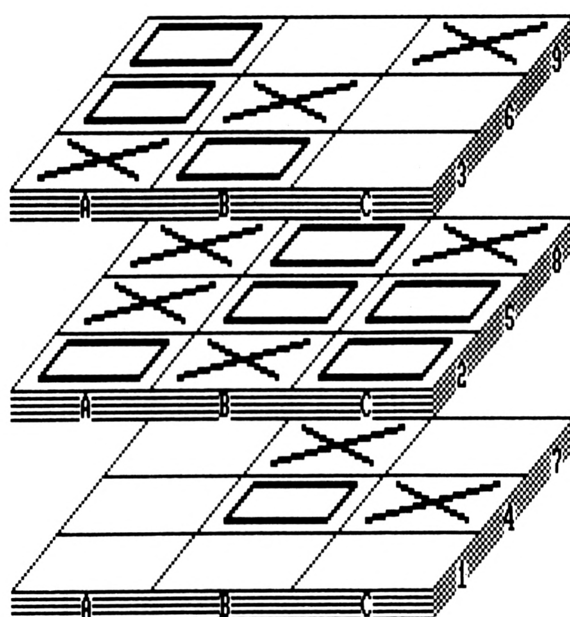
Tic Tac Toe en trois dimensions

Dans un échiquier en trois dimensions de trois fois trois fois trois cases, le but est d'aligner trois pions de la même couleur, en lignes, colonnes ou diagonales.




```
C'est a vous de commencer
Ou jouez-vous ? b5
Je joue en A1
Ou jouez-vous ? a8
Je joue en C2
Ou jouez-vous ? a2
Vous avez un coup gagnant
Quel que soit le mien
Je joue donc au hasard
Je joue en B1
Ou jouez-vous ? c8
Vous avez gagne
```

C'est la première partie que joue le programme : ses coups sont aléatoires, sa défaite est rapide.




```
C'est a moi de commencer
Je joue en B5
Ou jouez-vous ? a5
Je joue en C5
Ou jouez-vous ? a8
Je joue en A2
Ou jouez-vous ? c8
Je joue en B8
Ou jouez-vous ? b2
Je joue en B4
Ou jouez-vous ? b6
Je joue en B3
Ou jouez-vous ? b7
Je joue en A6
Ou jouez-vous ? c4
Je joue en C2
Ou jouez-vous ? a3
Vous avez un coup gagnant
Quel que soit le mien
Je joue donc au hasard
Je joue en A9
Ou jouez-vous ? c9
Vous avez gagne
```

Deuxième partie : le niveau du programme est nettement plus élevé, la victoire de son adversaire humain est longue à se dessiner.

```
-------------------------
 valeur des differentes
combinaisons de diagonales
-------------------------
..O=     -8 pts  ..X=     8 pts
.O.=      1 pts  .OO=    -1 pts
.OX=     -3 pts  .X.=    -1 pts
.XO=      3 pts  .XX=     1 pts
O..=     -8 pts  O.O=     3 pts
O.X=      0 pts  OO.=    -1 pts
OOO=  75002 pts  OOX=    -7 pts
OX.=      3 pts  OXO=     0 pts
OXX=      7 pts  X..=     8 pts
X.O=      0 pts  X.X=    -3 pts
XO.=     -3 pts  XOO=    -7 pts
XOX=      0 pts  XX.=     1 pts
XXO=      7 pts  XXX=-75002 pts
-------------------------
Appuyez sur ENTER pour refaire
 une partie
```

Après la deuxième partie, le programme affiche les valeurs des 27 combinaisons de triplets. Certaines d'entre elles auront besoin d'être affinées au cours des parties suivantes : 00. à une valeur sous-évaluée car cette combinaison mène souvent à la victoire.

Comment utiliser ces deux logiciels

L'utilisation est des plus simples. Lorsque le programme demande "Où jouez-vous", l'utilisateur n'a qu'à rentrer son coup, sous la forme d'une lettre suivie d'un chiffre (exemple B4). Tout le reste est géré automatiquement. Chacun commence à tour de rôle et, à la fin de chaque partie, le programme modifie certaines variables, de façon à assimiler l'expérience de son adversaire. Il visualise ensuite ces variables en affichant toutes les combinaisons de triplets de pions (pour Tic Tac Toe) et quadruplets (pour puissance 4), accompagnées de leurs nouvelles valeurs.

COMMENT ÇA MARCHE ?

La technique la plus utilisée, et jusqu'à maintenant la plus efficace, en programmation des jeux de stratégie, est le procédé minimax alpha-bêta. Bien que cela ne soit pas l'objet de cet ouvrage, il est utile de revenir sur ce principe.

Le procédé minimax est associé à l'exploration de l'arbre des différentes séries de coups qui pourraient intervenir durant la suite d'une partie. Cet arbre est considéré à partir de l'état de la partie au moment où l'ordinateur doit choisir un coup.

A chaque étape de l'exploration de cet arbre, l'ordinateur doit évaluer la position engendrée. Ainsi une note positive très importante signifiera que la position est excellente pour la machine. A l'inverse, une note très négative correspondra à une excellente position pour l'adversaire de la machine. Cette évaluation de la position est effectuée, très logiquement, par la fonction d'évaluation : une fonction qui dépend du jeu, donc qui représente, règles à part, la seule connaissance qu'a l'ordinateur de ce jeu.

Dans l'arbre d'exploration, parmi un groupe de coups entre lesquels devraient choisir son adversaire, l'ordinateur supposera que le coup choisi serait celui qui minimise la valeur de la position de la machine. Et inversement l'ordinateur choisirait, parmi un groupe de coups, celui qui maximise son propre score. Tout cela est mis au conditionnel car cette exploration n'est qu'une simulation.

Le plus difficile à réaliser n'est pas le programme d'exploration, mais plutôt celui d'évaluation d'une position. Celui-ci se construit en effet de manière très empirique car il intègre les très nombreux paramètres qui permettent de juger de l'intérêt d'une position. Le programmeur doit décider de leur importance relative, importance qui peut varier suivant les cas. De nombreux essais sont nécessaires, et finalement la recherche vers la perfection ne s'arrête jamais.

C'est là tout l'intérêt des programmes PUISSANCE 4 et TIC TAC TOE 3D que nous vous proposons. Ils décident du coup à jouer par une classique exploration d'arbre (minimax alpha-bêta) d'une profondeur de deux niveaux, mais l'originalité réside dans le fait que la fonction d'évaluation se construit "toute seule" au fur et à mesure des parties, en tenant compte de ce que joue son adversaire humain.

En d'autres termes, ces deux programmes puisent dans l'expérience de leur adversaire, et se perfectionnent peu à peu, à condition que celui-ci ne joue pas "n'importe quoi".

Au jeu puissance 4, le but est d'aligner 4 pions, en ligne, colonne, ou diagonale. Au jeu du Tic Tac Toe en 3 dimensions, les règles sont les mêmes, à ceci près qu'il s'agit d'aligner 3 pions dans un environnement tridimensionnel. Sur le plan du fonctionnement, les deux programmes sont assez identiques, prenons donc l'exemple du programme de Puissance 4. Il suppose que le damier est composé d'un ensemble de quadruplets, un quadruplet étant un groupe de 4 cases successives.

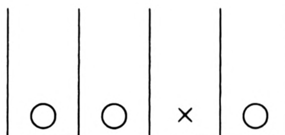
Figure 1

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

Les quadruplets sont les groupes de quatre cases.
Quelques exemples de quadruplets :
(1, 2, 3, 4) ou encore (1, 9, 17, 25) ou (18, 25, 32, 39).
Les quadruplets qui passent par la case 9 sont :
(1, 9, 17, 25) ; (2, 9, 16, 23) ; (8, 9, 10, 11) mais aussi (9, 10, 11, 12) ; (9, 16, 23, 30) ; (9, 17, 25, 33).

Chaque quadruplet est représenté en mémoire par un nombre virtuellement en base 3, en fait converti en base 10 car le Basic ne connaît pas la base 3. Chacun des quatre chiffres de ce nombre en base 3 est la représentation de l'état d'une des 4 cases du quadruplet : 0 correspond à une case vide, 1 à un pion noir, 2 à un pion blanc, 1, 2 et 3 étant les seuls chiffres autorisés en base 3.

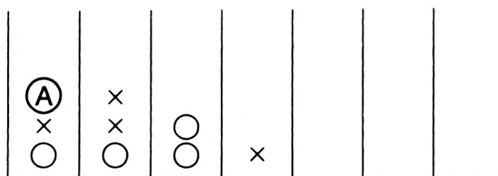
Figure 2



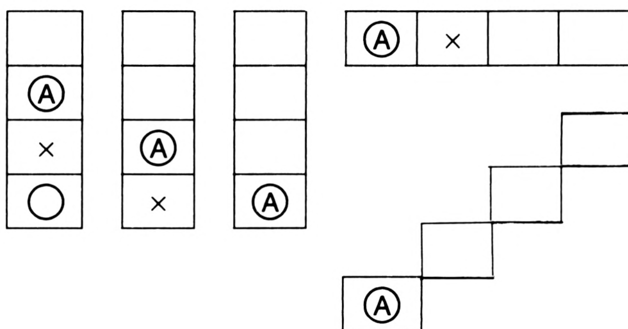
L'état de chaque quadruplet est mémorisé sous la forme d'un nombre en base 3, convertit en base 10. Si les ronds sont codés par le chiffre 1 et les croix par 2, ce nombre en base 3 est :
1121 soit en base 10 :
27 + 9 + 2*3 + 1 = 43

Au cours de chaque partie, le programme mémorise les quadruplets engrendrés par les coups de son adversaire et par ses propres coups.

Figure 3



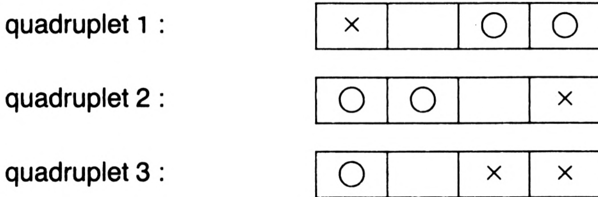
On entend par quadruplets engendrés en un coup, l'état de tous les quadruplets modifiés par ce coup. Ainsi les quadruplets engendrés par le coup A sont au nombre de 5.



En fin de partie, si son adversaire est gagnant, le programme incrémente les notes des quadruplets engendrés par le vainqueur et décrémente ceux engendrés par ses propres coups. Il fait de même pour les quadruplets symétriques, et fait

l'inverse pour les quadruplets... inverses, c'est-à-dire identiques si l'on remplace les pions noirs par des blancs et inversement.

Figure 4



Le quadruplet 2 est le symétrique du quadruplet 1
La quadruplet 3 est l'inverse du quadruplet 1.

Ainsi, par petites touches successives, la note, donc la mesure de l'intérêt de chaque quadruplet est affinée.

Lors de l'évaluation d'une position, le programme effectue la somme des notes de chaque type de quadruplet engendré par le coup qui vient d'être simulé.

A la fin de chaque partie, l'ordinateur affiche les valeurs des différents types de quadruplets.

On peut alors remarquer qu'un quadruplet de quatre pions identiques a soit une valeur très élevée (+75 000), soit une valeur très faible (−75 000) ; ces valeurs arbitrairement fixées au départ informent le programme que la position correspond respectivement à une victoire ou à une défaite. De même un quadruplet de deux pions identiques suivis de deux vides aura généralement une bonne note, car il est souvent joué par un adversaire intelligent qui sait qu'en multipliant cette combinaison il augmente ses chances de victoire.

Si son adversaire joue bien, l'ordinateur atteint en quelques parties un excellent niveau de jeu.

LISTE DES PRINCIPALES VARIABLES

pdiag(i,j)	case j du quadruplet i
case(i)	état de la case i (0, 1 ou 2)
diag(i,j)	numéro du i ème quadruplet auquel appartient la case i
ndiag(i)	nombre de quadruplets auquel appartient la case i
hdiag(i)	état du quadruplet i (nombre virtuellement en base 3)
coeff(i,j)	coefficient par lequel il faut multiplier l'état de la case j en vue d'ajouter le résultat à hdiag-i) lorsqu'un coup ayant modifié l'état de la diagonale i a été joué. Cela revient en fait à modifier un des chiffres en base 3 du nombre hdiag-i).
mem.diag(i,j)	j ième quadruplet engendré durant la dernière partie par le camp i (1 ou 2).
valeur(i)	valeur tactique du quadruplet i, c'est ce tableau qui est modifié après chaque partie.

```
10 REM*************************
20 REM  PUISSANCE 4 autodidacte
30 REM       T. L.-A.
40 REM*************************
50 MODE 2
60 PRINT"Patientez une minute SVP ..."
70 REM===============
80 REM initialisation
90 REM===============
100 DEFINT a-z
110 DEFREAL v
120 machine=1:humain=2:vmachine=1:vhumain=2:match.nul=3
130 DIM pdiag(69,3),case(41),casec(8,6)
140 DIM diag(12,41),coeff(12,41),ndiag(41)
150 DIM hdiag(68),mem.diag(2,200),diagm(12)
160 DIM inv(80),sym(80),valeur(80),pion$(2),petit.pion$(2)
170 valeur(40)=75000:valeur(80)=-75000
180 petit.pion$(0)=".":petit.pion$(1)=CHR$(233):petit.pion$(2)=CHR$(232)
190 SYMBOL 255,60,60,60,60,60,60,60,60
200 SYMBOL 254,255,255,255,255,170,85,170,85
210 FOR i=35 TO 41
220  c(i-35)=i
230 NEXT i
240 REM--- inverse et symetrique des diagonales ---
250 FOR i0=0 TO 2:i(0)=i0
260   FOR i1=0 TO 2:i(1)=i1
270     FOR i2=0 TO 2:i(2)=i2
280       FOR i3=0 TO 2:i(3)=i3
290         REM--- inverse ---
300         FOR j=0 TO 3
310           IF i(j)>0 THEN i1(j)=3-i(j) ELSE i1(j)=0
320         NEXT j
330         d=27*i(3)+9*i(2)+3*i(1)+i(0)
340         invd=27*i1(3)+9*i1(2)+3*i1(1)+i1(0)
350         inv(d)=invd
360         REM--- symetrique ---
370         d2=27*i(0)+9*i(1)+3*i(2)+i(3)
380         IF d<>d2 THEN sym(d)=d2:sym(d2)=d
390       NEXT i3
400     NEXT i2
410   NEXT i1
420 NEXT i0
430 REM--- dessin des pions ---
440 DATA 140,140,140,140,140,140,140,10,8,8,8,8,8,8,8
450 DATA 143,143,254,254,254,143,143,10,8,8,8,8,8,8,8
460 DATA 143,143,207,207,207,143,143,10,8,8,8,8,8,8,8
470 DATA 143,143,143,143,143,143,143
480 DATA 140,140,140,140,140,140,140,10,8,8,8,8,8,8,8
490 DATA 143,32,32,32,32,32,143,10,8,8,8,8,8,8,8
```

```
500 DATA 143,32,32,32,32,32,143,10,8,8,8,8,8,8,8
510 DATA 143,140,140,140,140,140,143
520 FOR camp=machine TO humain
530   FOR i=1 TO 52
540     READ c:c$=CHR$(c)
550     pion$(camp)=pion$(camp)+c$
560   NEXT i
570 NEXT camp
580 GOSUB 900
590 REM=================
600 REM affichage du jeu
610 REM=================
620 MODE 2
630 WINDOW 59,80,1,25
640 LOCATE#1,1,25:PRINT#1,STRING$(57,207);
650 FOR x=1 TO 57 STEP 8
660   FOR y=1 TO 25
670     LOCATE#1,x,y:PRINT#1,CHR$(255);
680   NEXT y
690 NEXT x
700 FOR x=5 TO 53 STEP 8
710   LOCATE#1,x,25:PRINT#1,RIGHT$(STR$(INT(x/8+1)),1);
720 NEXT x
730 GOTO 1340
740 REM========================================
750 REM affiche le pion (humain ou machine) en p
760 REM========================================
770 y=INT(p/7):x=p-7*y
780 x=2+8*x:y=4*y+1
790 FOR ym=1 TO y
800   IF ym>1 THEN LOCATE#1,x,ym-1:PRINT#1,STRING$(7,32);
810   LOCATE#1,x,ym
820   FOR ii=1 TO LEN (pion$(camp)) STEP 10
830     PRINT#1,MID$(pion$(camp),ii,10);
840   NEXT ii
850 NEXT ym
860 RETURN
870 REM============================================================
880 REM reperage de la position de chaque case dans chaque diagonale
890 REM============================================================
900 FOR x=0 TO 8 STEP 8
910   FOR y=0 TO 6
920     casec(x,y)=1
930   NEXT y
940 NEXT x
950 REM---
960 FOR x=0 TO 8
970   casec(x,6)=1
980 NEXT x
```

```
990 REM---
1000 FOR x=1 TO 7
1010   FOR y=0 TO 6
1020     FOR dx=-1 TO 1
1030       FOR dy=0 TO 1
1040         IF dx=0 AND dy=0 THEN 1130
1050         IF dx=-1 AND dy=0 THEN 1130
1060         xm=x:ym=y
1070         p=7*ym+xm-1
1080         pdiag(id,oc)=p
1090         IF casec(xm,ym)=1 THEN 1120
1100         IF oc=3 THEN id=id+1:GOTO 1120
1110         oc=oc+1:xm=xm+dx:ym=ym+dy:GOTO 1070
1120         oc=0
1130       NEXT dy
1140     NEXT dx
1150   NEXT y
1160 NEXT x
1170 REM---
1180 FOR p=0 TO 41
1190   i=0
1200   FOR id=0 TO 68
1210     FOR oc=0 TO 3
1220       IF pdiag(id,oc)<>p THEN 1260
1230       diag(i,p)=id
1240       coeff(i,p)=3^oc
1250       i=i+1
1260     NEXT oc
1270     ndiag(p)=i-1
1280   NEXT id
1290 NEXT p
1300 RETURN
1310 REM================
1320 REM debut de partie
1330 REM================
1340 IF comm=1 THEN comm=0 ELSE comm=1
1350 IF comm=1 THEN PRINT "Vous commencez":GOTO 1400
1360 IF comm=0 THEN PRINT "Je commence":GOTO 1530
1370 REM=====================
1380 REM entree du coup humain
1390 REM=====================
1400 INPUT "Ou jouez-vous ";col:col=col-1
1410 IF col<0 OR col>6 THEN PRINT "coordonnees invalides":GOTO 1400
1420 IF c(col)<0 THEN PRINT "Colonne pleine":GOTO 1400
1430 camp=humain:p=c(col)
1440 GOSUB 2110
1450 GOTO 1470
1460 REM---- fin de partie ? ---
1470 IF fin<>non THEN 2270
```

```
1480 REM===================================
1490 REM recherche du meilleur coup machine
1500 REM par exploration minimax alpha-beta
1510 REM    2 niveaux de profondeur
1520 REM===================================
1530 val0=-100000
1540 REM--- exploration du niveau 1 ---
1550 FOR p0=0 TO 6
1560   IF c(p0)<0 THEN 1880
1570   REM--- simulation et evaluation au niveau 1 ---
1580   val2=0
1590   FOR i=0 TO ndiag(c(p0))
1600     anc.diag=hdiag(diag(i,c(p0)))
1610     nouv.diag=anc.diag+coeff(i,c(p0))
1620     val2=val2-valeur(anc.diag)+valeur(nouv.diag)
1630     hdiag(diag(i,c(p0)))=nouv.diag
1640     diagm(i)=anc.diag
1650   NEXT i
1660   c(p0)=c(p0)-7
1670   REM--------------------------------------------
1680   valm=val2
1690   IF val2>50000 THEN meilleur.coup=p0:GOSUB 1910:GOTO 1970
1700   val1=100000
1710   REM--- exploration du niveau 2 ---
1720   FOR p1=0 TO 6
1730     IF c(p1)<0 THEN 1840
1740     REM--- simulation et evaluation au niveau 2 ---
1750     val2=valm
1760     FOR i=0 TO ndiag(c(p1))
1770       anc.diag=hdiag(diag(i,c(p1)))
1780       nouv.diag=anc.diag+2*coeff(i,c(p1))
1790       val2=val2-valeur(anc.diag)+valeur(nouv.diag)
1800     NEXT i
1810     REM--------------------------------------------
1820     IF val2<=val0 THEN 1870
1830     IF val2<val1 THEN val1=val2
1840   NEXT p1
1850   IF val1>val0 THEN val0=val1:meilleur.coup=p0
1860   REM--- remontee au niveau 1 ---
1870   GOSUB 1910
1880 NEXT p0
1890 GOTO 1970
1900 REM--- remontee au niveau 0 ---
1910 c(p0)=c(p0)+7
1920 FOR i=0 TO ndiag(c(p0))
1930   hdiag(diag(i,c(p0)))=diagm(i)
1940 NEXT i
1950 RETURN
1960 REM----------------------------
```

```
1970 p=c(meilleur.coup):camp=machine:col=meilleur.coup
1980 GOSUB 2110
1990 IF NOT(val0<-40000 AND fin<>vmachine) THEN 2030
2000 PRINT "Vous avez un coup"
2010 PRINT "gagnant quel que"
2020 PRINT "soit le mien"
2030 PRINT "Je joue en ";col+1
2040 IF fin<>0 THEN 2270
2050 GOTO 1400
2060 REM========================================
2070 REM modification du jeu et memorisation des
2080 REM        diagonales engendrees
2090 REM         entree : camp  et p
2100 REM========================================
2110 FOR i=0 TO ndiag(p)
2120   d=diag(i,p)
2130   hdiag(d)=hdiag(d)+coeff(i,p)*camp
2140   mem.diag(camp,cij(camp))=hdiag(d)
2150   cij(camp)=cij(camp)+1
2160   IF hdiag(d)=40 THEN fin=vmachine:diag.vict=d
2170   IF hdiag(d)=80 THEN fin=vhumain:diag.vict=d
2180 NEXT i
2190 GOSUB 770
2200 c(col)=c(col)-7
2210 ncases.pleines=ncases.pleines+1
2220 IF ncases.pleines=42 THEN fin=match.nul
2230 RETURN
2240 REM=======================
2250 REM annonce de fin de partie
2260 REM=======================
2270 IF fin=match.nul THEN PRINT "Math nul":GOTO 2540
2280 IF fin=vmachine THEN PRINT "J'ai gagne"
2290 IF fin=vhumain THEN PRINT "Vous avez gagne"
2300 REM===============================
2310 REM modification des valeurs des
2320 REM diagonales jouees par le gagnant
2330 REM durant la derniere partie
2340 REM===============================
2350 gagnant=fin:perdant=3-fin
2360 IF gagnant=machine THEN 2540
2370 camp=gagnant:GOSUB 2410
2380 camp=perdant:GOSUB 2410
2390 GOTO 2540
2400 REM---------
2410 FOR i=0 TO cij(camp)-1
2420   mem=mem.diag(camp,i)
2430   valeur(inv(mem))=valeur(inv(mem))+1
2440   mem1=sym(inv(mem))
2450   IF mem1<>0 THEN valeur(mem1)=valeur(mem1)+1
```

```
2460   valeur(mem)=valeur(mem)-1
2470   mem2=sym(mem)
2480   IF mem2<>0 THEN valeur(mem2)=valeur(mem2)-1
2490 NEXT i
2500 RETURN
2510 REM=================
2520 REM re-initialisation
2530 REM=================
2540 FOR i=35 TO 41
2550  c(i-35)=i
2560 NEXT i
2570 FOR d=0 TO 68
2580   hdiag(d)=0
2590 NEXT d
2600 fin=non
2610 cij(1)=0:cij(2)=0
2620 GOSUB 2700
2630 PRINT  "Appuyez sur ENTER"
2640 PRINT "pour refaire":INPUT "une partie",rep$
2650 ncases.pleines=0
2660 GOTO 620
2670 REM===================================
2680 REM affichage des valeurs des diagonales
2690 REM===================================
2700 PRINT "--------------------"
2710 PRINT"valeur des":PRINT"combinaisons de":PRINT"quadruplets"
2720 PRINT "--------------------"
2730 FOR i=1 TO 80
2740   i1=i
2750   d(0)=INT(i1/27):d(1)=INT((i1-27*d(0))/9)
2760   d(2)=INT((i1-9*d(1)-27*d(0))/3):d(3)=i1-27*d(0)-9*d(1)-3*d(2)
2770   FOR j=0 TO 3
2780     PRINT petit.pion$(d(j));
2790   NEXT j
2800   PRINT "=";USING "######";valeur(i);
2810   PRINT
2820 NEXT i
2830 PRINT "--------------------"
2840 RETURN
2850 w$=INKEY$:IF w$="" THEN 2850
2860 IF w$=" " THEN RETURN
2870 SAVE"d"+CHR$(oip+48)+".LOR",b,49152,16384:oip=oip+1:RETURN
```

```
10 REM***************************************
20 REM  TIC TAC TOE 3 DIMENSIONS autodidacte
30 REM            T. L.-A.
40 REM***************************************
50 PLOT 1,1,1
60 MODE 2
70 PRINT"Patientez une minute SVP ..."
80 REM================
90 REM initialisation
100 REM===============
110 DEFINT a-z:DEFREAL v
120 machine=1:vmachine=1
130 humain=2:vhumain=2
140 match.nul=3
150 DIM pdiag(48,2),hdiag(48)
160 DIM diag(13,26),coeff(13,26),ndiag(26),case(26),valeur(26)
170 DIM cij(2),mem.diag(2,200),inv(26),sym(26),diagm(13)
180 DIM piece$(2)
190 valeur(13)=75000:valeur(26)=-75000
200 piece$(0)=".":piece$(1)="O":piece$(2)="X"
210 REM--- inverse des diagonales (XXO est l'inverse de OOX) ---
220 DATA 0,2,1,6,8,7,3,5,4,18
230 DATA 20,19,24,26,25,21,23,22
240 DATA 9,11,10,15,17,16,12,14,13
250 FOR i=0 TO 26
260   READ inv(i)
270 NEXT i
280 REM--- symetrique des diagonales (XXO est le symetrique de OXX) ---
290 DATA 4,12,2,18,1,9,5,21,7,15,8,24,11,19,14,22,17,25
300 FOR i=1 TO 9
310   READ a,b
320   sym(a)=b:sym(b)=a
330 NEXT i
340 GOSUB 990
350 REM=================
360 REM affichage du jeu
370 REM=================
380 MODE 2:WINDOW 50,79,2,24
390 FOR y=32 TO 256 STEP 112
400    FOR dy=2 TO 14 STEP 4
410       MOVE 0,y-dy-1:DRAW 288,y-dy-1:DRAW 384,y-dy+95
420    NEXT dy
430    FOR x=0 TO 288 STEP 96
440       MOVE x,y
450       DRAW x+96,y+96
460    NEXT x
470    FOR d=0 TO 96 STEP 32
480       MOVE d,y+d
490       DRAW d+288,y+d
```

```
500    NEXT d
510 NEXT y
520 MOVE 390,15:DRAW 390,390:DRAW 639,390
530 DRAW 639,15:DRAW 390,15
540 FOR x=7 TO 31 STEP 12
550   FOR y=10 TO 24 STEP 7
560     LOCATE#1,x,y:PRINT#1,CHR$(65+(x-7)/12);
570   NEXT y
580 NEXT x
590 i=8
600 FOR d=0 TO 8 STEP 4
610    FOR y=5 TO 19 STEP 7
620       LOCATE#1,47-d,y+d/2:PRINT#1,CHR$(49+i);:i=i-1
630    NEXT y
640 NEXT d
650 GOTO 1200
660 REM============================================
670 REM affichage du pion (carre ou croix) en x,y,z
680 REM============================================
690 xp=64+96*x+32*z
700 xp=64+96*x+32*z
710 yp=48+112*y+32*z
720 IF camp=machine THEN 750
730 IF camp=humain THEN 810
740 REM--- affichage d'un carre ---
750 FOR dy=0 TO 2
760   MOVE xp+45,yp+10+dy:DRAW xp-24,yp+10+dy
770   DRAW xp-45,yp-10+dy:DRAW xp+24,yp-10+dy:DRAW xp+45,yp+10+dy
780 NEXT dy
790 RETURN
800 REM--- affichage d'une croix ---
810 FOR dy=0 TO 2
820   MOVE xp+45,yp+10+dy:DRAW xp-45,yp-10+dy
830   MOVE xp+24,yp-10+dy:DRAW xp-24,yp+10+dy
840 NEXT dy
850 RETURN
860 REM===========================================================
870 REM reperage de la position de chaque case dans chaque diagonale
880 REM===========================================================
890 DATA 1,10,19,1,4,7,1,13,25,1,2,3,1,11,21
900 DATA 1,5,9,1,14,27,10,13,16,10,11,12,10,14,18
910 DATA 19,13,7,19,22,25,19,11,3,19,20,21,19,14,9
920 DATA 19,23,27,4,13,22,4,5,6,4,14,24,13,14,15
930 DATA 22,14,6,22,23,24,7,16,25,7,5,3,7,14,21
940 DATA 7,8,9,7,17,27,16,14,12,16,17,18,25,14,3
950 DATA 25,23,21,25,17,9,25,26,27,2,11,20,2,5,8
960 DATA 2,14,26,11,14,17,20,14,8,20,23,26,5,14,23
970 DATA 8,17,26,3,12,21,3,6,9,3,15,27,12,15,18
980 DATA 21,15,9,21,24,27,6,15,24,9,18,27
```

```
990 FOR id=0 TO 48
1000   FOR oc=0 TO 2
1010     READ a:pdiag(id,oc)=a-1
1020   NEXT oc
1030 NEXT id
1040 FOR p=0 TO 26
1050   i=0
1060   FOR id=0 TO 48
1070     FOR oc=0 TO 2
1080       IF pdiag(id,oc)<>p THEN 1120
1090       diag(i,p)=id
1100       coeff(i,p)=3^oc
1110       i=i+1
1120     NEXT oc
1130     ndiag(p)=i-1
1140   NEXT id
1150 NEXT p
1160 RETURN
1170 REM================
1180 REM debut de partie
1190 REM================
1200 IF comm=1 THEN comm=0 ELSE comm=1
1210 IF comm=1 THEN PRINT "C'est a vous de commencer":GOTO 1260
1220 IF comm=0 THEN PRINT "C'est a moi de commencer":GOTO 1450
1230 REM=====================
1240 REM entree du coup humain
1250 REM=====================
1260 INPUT "Ou jouez-vous ";rep$
1270 IF LEN(rep$)<>2 THEN 1370
1280 x=ASC(UPPER$(LEFT$(rep$,1)))-65
1290 yz=VAL(RIGHT$(rep$,1))-1
1300 IF x<0 OR x>2 THEN 1370
1310 IF yz<0 OR yz>8 THEN 1370
1320 p=x+3*yz
1330 IF case(p)<>0 THEN PRINT "Case occupee":GOTO 1260
1340 camp=humain
1350 GOSUB 2020
1360 GOTO 1390
1370 PRINT "coordonnees invalides":GOTO 1260
1380 REM---- fin de partie ? ---
1390 IF fin<>non THEN 2190
1400 REM===================================
1410 REM recherche du meilleur coup machine
1420 REM par exploration minimax alpha-beta
1430 REM    2 niveaux de profondeur
1440 REM===================================
1450 val0=-100000
1460 REM--- exploration du niveau 1 ---
1470 FOR p0=0 TO 26
```

```
1480   IF case(p0)<>0 THEN 1800
1490   REM--- simulation et evaluation au niveau 1 ---
1500   val2=0
1510   FOR i=0 TO ndiag(p0)
1520     anc.diag=hdiag(diag(i,p0))
1530     nouv.diag=anc.diag+coeff(i,p0)
1540     val2=val2-valeur(anc.diag)+valeur(nouv.diag)
1550     hdiag(diag(i,p0))=nouv.diag
1560     diagm(i)=anc.diag
1570   NEXT i
1580   case(p0)=machine
1590   REM-------------------------------------------
1600   valm=val2
1610   IF val2>50000 THEN meilleur.coup=p0:GOSUB 1830:GOTO 1880
1620   val1=100000
1630   REM--- exploration du niveau 2 ---
1640   FOR p1=0 TO 26
1650     IF case(p1)<>0 THEN 1760
1660     REM--- simulation et evaluation au niveau 2 ---
1670     val2=valm
1680     FOR i=0 TO ndiag(p1)
1690       anc.diag=hdiag(diag(i,p1))
1700       nouv.diag=anc.diag+2*coeff(i,p1)
1710       val2=val2-valeur(anc.diag)+valeur(nouv.diag)
1720     NEXT i
1730     REM-------------------------------------------
1740     IF val2<=val0 THEN 1790
1750     IF val2<val1 THEN val1=val2
1760   NEXT p1
1770   IF val1>val0 THEN val0=val1:meilleur.coup=p0
1780   REM--- remontee au niveau 1 ---
1790   GOSUB 1830
1800 NEXT p0
1810 GOTO 1880
1820 REM--- remontee au niveau 0 ---
1830 case(p0)=0
1840 FOR i=0 TO ndiag(p0)
1850   hdiag(diag(i,p0))=diagm(i)
1860 NEXT i
1870 RETURN
1880 p=meilleur.coup:camp=machine
1890 GOSUB 2020
1900 IF NOT(val0<-40000 AND fin<>vmachine) THEN 1940
1910 PRINT "Vous avez un coup gagnant"
1920 PRINT "Quel que soit le mien"
1930 PRINT "Je joue donc au hasard"
1940 PRINT "Je joue en ";CHR$(65+x);RIGHT$(STR$(1+y+3*z),1)
1950 IF fin<>non THEN 2190
1960 GOTO 1260
```

```
1970 REM=============================================
1980 REM modification du jeu et memorisation des
1990 REM        diagonales engendrees
2000 REM         entree : camp  et p
2010 REM=============================================
2020 FOR i=0 TO ndiag(p)
2030   d=diag(i,p)
2040   hdiag(d)=hdiag(d)+coeff(i,p)*camp
2050   mem.diag(camp,cij(camp))=hdiag(d)
2060   cij(camp)=cij(camp)+1
2070   IF hdiag(d)=13 THEN fin=vmachine:diag.vict=d
2080   IF hdiag(d)=26 THEN fin=vhumain:diag.vict=d
2090 NEXT i
2100 case(p)=camp
2110 z=INT(p/9):y=INT((p-9*z)/3):x=p-9*z-3*y
2120 GOSUB 690
2130 ncases.pleines=ncases.pleines+1
2140 IF ncases.pleines=27 THEN fin=match.nul
2150 RETURN
2160 REM=========================
2170 REM annonce de fin de partie
2180 REM=========================
2190 IF fin=match.nul THEN PRINT "Math nul":GOTO 2460
2200 IF fin=vmachine THEN PRINT "J'ai gagne"
2210 IF fin=vhumain THEN PRINT "Vous avez gagne"
2220 REM=================================
2230 REM modification des valeurs des
2240 REM diagonales jouees par le gagnant
2250 REM durant la derniere partie
2260 REM=================================
2270 gagnant=fin:perdant=3-fin
2280 IF gagnant=machine THEN 2460
2290 camp=gagnant:GOSUB 2330
2300 camp=perdant:GOSUB 2330
2310 GOTO 2460
2320 REM---------
2330 FOR i=0 TO cij(camp)-1
2340   mem=mem.diag(camp,i)
2350   valeur(inv(mem))=valeur(inv(mem))+1
2360   mem1=sym(inv(mem))
2370   IF mem1<>0 THEN valeur(mem1)=valeur(mem1)+1
2380   valeur(mem)=valeur(mem)-1
2390   mem2=sym(mem)
2400   IF mem2<>0 THEN valeur(mem2)=valeur(mem2)-1
2410 NEXT i
2420 RETURN
2430 REM==================
2440 REM re-initialisation
2450 REM==================
```

```
2460 FOR p=0 TO 26
2470   case(p)=0
2480 NEXT p
2490 FOR d=0 TO 48
2500   hdiag(d)=0
2510 NEXT d
2520 fin=non
2530 cij(1)=0:cij(2)=0
2540 GOSUB 2610
2550 INPUT "Appuyez sur ENTER pour refaire une partie",rep$
2560 ncases.pleines=0
2570 GOTO 380
2580 REM=====================================
2590 REM affichage des valeurs des diagonales
2600 REM=====================================
2610 PRINT "--------------------------"
2620 PRINT " valeur des differentes"
2630 PRINT "combinaisons de triplets"
2640 PRINT "--------------------------"
2650 FOR i=1 TO 26
2660   i1=i
2670   c(0)=INT(i1/9):c(1)=INT((i1-9*c(0))/3):c(2)=i1-3*c(1)-9*c(0)
2680   FOR j=0 TO 2
2690     PRINT piece$(c(j));
2700   NEXT j
2710   PRINT "=";USING "######";valeur(i);:PRINT " pts";
2720  IF INT(i/2)<>i/2 THEN PRINT "  ";
2730 NEXT i
2740 PRINT "--------------------------"
2750 RETURN
```

CHAPITRE 4
LES SYSTÈMES EXPERTS

DES APPLICATIONS RENTABLES

Les systèmes experts (SE) constituent un des rares domaines de l'intelligence artificielle réellement opérationnels. Avec les disciplines liées aux fonctions de la perception, nous abordions certains, parmi les plus prometteurs, des secteurs de la recherche en intelligence artificielle. Les systèmes experts tiennent, eux, déjà leurs promesses. Grâce à eux, l'intelligence artificielle a acquis ses lettres de noblesse dans les laboratoires les plus importants, non seulement publics, mais aussi privés (IBM, ICL, Schlumberger, SRI International, etc.). Ce dernier point est très significatif : ces laboratoires ne s'intéresseraient probablement pas à un secteur non rentable à court terme : non rentable n'est certainement pas le qualificatif à appliquer aux systèmes experts.

Ces systèmes font désormais partie de la panoplie des instruments les plus sophistiqués des spécialistes de tout poil. Comme son nom l'indique, un système expert simule le raisonnement d'un expert. Les domaines couverts peuvent aller de la médecine à la détection de pannes en passant par la prospection géologique, la chimie etc. c'est-à-dire tous ceux nécessitant un avis d'expert. Or, la différence fondamentale entre un expert humain et un système expert fait rêver tous les PDG des grandes entreprises.

Quelques systèmes experts

Nom du SE	Auteurs	Sujet
Casnet	Weiss et Kulikowski (U. de Rutgers)	Médecine (Glaucome de l'œil)
Centaur	Aikins (U. de Stanford)	Médecine
Crib	Addis (ICL)	Détection de fautes
Crysalis	Feigenbaum et Engelmore (Stanford)	Détermination de la structure de protéines à partir de densités

Nom du SE	Auteurs	Sujet
Dart	Université de Stanford & IBM	Engineering
Dentral	Lederberg	Science (1ier SE : 1965)
Dipmeter	Davis	Déterminatioin de couches géologiques
Emycin	Van Melle	Extended MYCIN. Engineering
Expert	Weiss et Kulikowski	Engineering
Gamma	Barstow U. de Yale	Science
Headmed	Heiser	Médecine
Internist	Pople	Médecine
Litho	Bonnet (B1) Schlumberger	Aide à la recherche pétrolière
Macsyma	Gensereth	Mathématiques
MDX	Chandrasekaran	Médecine, aide au diagnostic des maladies du cholestérol
Mycin	Shortliffe Stanford	Médecine, aide à la décision pour les infections bact. du sang
Oncocin	Shortliffe et Scott Stanford	Médecine, aide à la décision dans la chimiothérapie des leucémiques
Prospector	Konolige	Aide à la prospection minière
Raffles	Addis ICL	Détection de pannes
Rita	Anderson Rand corporation	Engineering
Simmias	Laurière et Perrot Schlumberger	Aide à la recherche pétrolière

Un système expert ne tombe jamais malade, il n'a pas d'états d'âme, il ne coûte plus un centime après la phase de développement, ne demande jamais d'augmentation ni ne menace de passer à la concurrence en cas de refus.

A côté de ces avantages, le point le plus important... et le plus dur à avaler : les systèmes experts égalent l'efficacité des experts qui leur ont servi de mentor. Contrairement à tous les autres systèmes d'intelligence artificielle, ceux-ci, et eux seuls, sont véritablement opérationnels. C'est dire leur enjeu économique et la méfiance qui accompagne toujours ce genre de révolution.

QU'EST-CE QUE C'EST ?

Comme nous l'avons dit précédemment, on distingue deux disciplines fondamentales en informatique : l'algorithmique et l'heuristique. Plus une troisième, la statistique qui intervient dans les deux précédentes. Pour être véritablement efficace, un système d'intelligence artificielle doit obligatoirement exploiter ces trois disciplines. La caractéristique principale d'un système expert (SE) tient au fait que l'heuristique est largement privilégiée par rapport à l'algorithmique et la statistique. Cela signifie que pour résoudre un problème, un système expert exploitera un ensemble de règles de raisonnement reposant sur un ensemble de faits, plutôt que de suivre un algorithme de résolution l'amenant directement à la solution recherchée.

Principal avantage de la démarche : contrairement à l'algorithmique, il n'est pas nécessaire de connaître a priori le chemin qui mène au résultat cherché. Ce chemin peut être déterminé en cours de recherche en simulant le raisonnement humain. Grâce à ce mode de fonctionnement, un système expert peut très bien s'appliquer à des sciences inexactes : expérimentales, humaines, etc. Tous domaines où le moyen de parvenir à une solution est aussi inconnue que la solution elle-même, où d'ailleurs ces moyens comme parfois les solutions sont multiples.

ALGORITHMIQUE ET HEURISTIQUE PAR L'EXEMPLE

Pour mieux comprendre les différences fondamentales entre approche algorithmique et heuristique d'un problème, appliquons successivement ces deux techniques dans la résolution d'un problème élémentaire : une équation du second degré.

Soit l'équation à résoudre : $aX + bX + c = 0$

C'est-à-dire trouver X tel que l'équation soit vraie, a, b et c étant connus.

Deux voies peuvent être envisagées pour parvenir à la solution : soit appliquer la "formule magique" apprise par cœur par l'écolier ou l'ordinateur, soit chercher cette solution par tâtonnements. La première solution relève de l'algorithmique, la seconde de l'heuristique.

L'ALGORITHMIQUE

On considère que l'équation $aX + bX + c = 0$ est de la forme $AX + BX + C = 0$, A, B, C étant quelconques. La formule magique à appliquer est $X = C/(A+B)$ si A diffère de B. On remplace alors A, B, C par a, b, c et on a $X = -c/(a+b)$.

Et cela sans effort de raisonnement, mais la mémoire est mise à rude épreuve et un problème un peu sérieux nécessitera la mémorisation de centaines de formules de ce type !

L'inconvénient majeur de cette méthode réside dans la trop grande spécialisation des algorithmes. Ainsi celui-ci, capable de résoudre l'équation $aX + bX + c = 0$ baisse les bras face à une équation aussi ressemblante que $aX + bX + cX + d = 0$.

On dit souvent que les ordinateurs sont bêtes au point qu'il faut tout leur préciser, jusqu'à la moindre addition à effectuer : c'est exact en ce qui concerne l'algorithmique. La grande révolution des systèmes experts et de l'heuristique en général tient justement au fait que, tout comme l'homme, le système se construit son propre algorithme en fonction du problème à traiter.

L'HEURISTIQUE

L'écolier qui ne désire pas apprendre par cœur toutes les formules existantes dispose d'une alternative autrement plus payante : trouver n'importe quelle formule par le raisonnement. La solution est alors construite pas à pas. Avantage : face à des problèmes différents mais de même type (deux équations algébriques, par exemple) les mêmes règles sont utilisées avec succès. Cette approche relève de l'heuristique et les systèmes experts en sont une application. Si l'on reprend notre exemple d'équation du premier degré, trois règles suffisent à retrouver la solution :

REGLE 1 : SI X+A=0 ALORS X=−A
REGLE 2 : SI AX+BX+C=0 ALORS (A+B)X+C=0
REGLE 3 : SI AX=B et A<>0 ALORS X=B/A

Le système expert applique alors ces règles sur notre exemple aX+bX+c=0 :

REGLE 2 ⇒ (a+b)X+c=0
REGLE 1 ⇒ (a+b)X=−c
REGLE 3 ⇒ si a+b<>0 ALORS X=−c/(a+b)

STRUCTURE D'UN SYSTÈME EXPERT

Toute l'intelligence artificielle gravite autours de ces deux pôles : la représentation de la connaissance d'une part, la démonstration automatique de l'autre. Les bases de données, les bases d'images appartiennent au premier pôle. La démonstration automatique de théorèmes ne relève que du second. Quand aux systèmes experts, ils se situent à la jonction des deux et constituent une des rares, pour ne pas dire la seule application opérationnelle qui réalise cette synthèse. Pour des raisons multiples, facilité de mise-à-jour des connaissances, efficacité, portabilité, les systèmes experts possèdent pour la plupart une structure modulaire, comportant deux parties :

– une base de connaissance,
– un démonstrateur (ou moteur d'inférence).

LA BASE DE CONNAISSANCE

Elle se compose de faits et de règles. Sa principale originalité par rapport à une base de données classique, est que les règles peuvent enrichir la base de faits, par l'intermédiaire du moteur d'inférence.

Considérons la base de connaissance suivante (exemple 1) :

1. Base de faits : (B, E)

2. Base de règles :

R1 : A et B ⇒ F
R2 : B et H ⇒ A
R3 : B et C et D ⇒ A
R4 : E ⇒ C
R5 : F ⇒ G
R6 : C ⇒ D
R7 : I ⇒ B

Les faits sont des assertions telles que "Socrate est un homme". Les règles mettent en relation certains faits entre eux. Les relations peuvent être de type très divers, la plus courante étant la relation de cause à effet (si... alors). Par exemple et pour reprendre le célèbre sophisme, une règle pourra être "si homme alors mortel". Dans l'exemple ci-dessus, la règle R1 signifie "si A et B sont vrais alors F est vrai".

LE MOTEUR D'INFÉRENCE

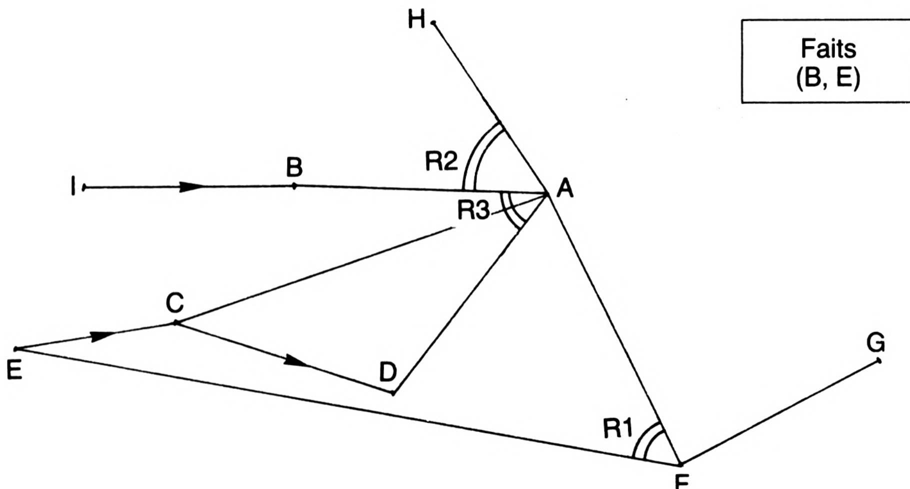
1. Chaînage avant

Partant de la base initiale, on applique les règles dont les prémisses sont satisfaites. Les faits ainsi déduits sont ajoutés à la base de faits. Reprenons l'exemple 1 et appliquons cette technique. A chaque règle appliquée, la base s'enrichit d'un nouveau fait.

Règle appliquée	Base de faits
	(B,E)
R4 : E ⇒ C	(B,C,E)
R6 : C ⇒ D	(B,C,D,E)
R3 : B ET C ET D ⇒ A	(A,B,C,D,E)
R1 : A ET B ⇒ F	(A,B,C,D,E,F)
R5 : F ⇒ G	(A,B,C,D,E,F,G)

2. Chaînage arrière

Plutôt que de tenter d'appliquer chaque règle à chaque fait, ce qui se révèle parfois long, on se concentre sur un but à atteindre, par exemple G. On examine alors toutes les règles qui ont G pour conséquence et pour chacune de ces règles, on regarde si les prémisses appartiennent à la base de faits. Si c'est le cas, le but est démontré, sinon on considère les prémisses comme autant de sous-buts à atteindre et on itère le processus, c'est-à-dire que l'on regarde les règles qui ont ces prémisses pour conséquence et ainsi de suite.



Règles

	A et E	→ F	(1)
	B et H	→ A	(2)
	B et C et D	→ A	(3)
*	E	→ C	(4)
	F	→ G	(5)
	G	→ D	(6)
	I	→ B	(7)

Ainsi dans l'exemple 1, et en s'aidant de l'arbre ci-après, on voit que G est conséquence de F (Règle R5). Or, F n'appartient pas à la base de faits et devient donc le nouveau fait à démontrer. Par R1, F est conséquence de A et B. Le fait B appartient à la base de faits, mais par le fait A, celui-ci devient donc le nouveau fait à démontrer. A est conséquence de deux règles : R3 et R2. Ici intervient la notion d'ordre d'application des règles. Dans les systèmes dits statiques, cet ordre est défini une bonne fois pour toutes. On appliquera donc R2, puis, en cas d'impasse, R3.

Dans des moteurs d'inférence plus sophistiqués, cet ordre peut dépendre du contexte. Par exemple on choisira la règle qui a le plus petit nombre de prémisses non satisfaites. Supposons notre système statique, et appliquons donc la règle 2 en priorité : les faits B et H en sont les prémisses. Or, H n'est conséquence d'aucune règle et n'est pas dans la base de faits. Il est donc impossible à démontrer et il faut répartir du but A, conséquence de la règle R3. Les prémisses sont B, C et D. Le fait B appartient à la base. Le fait C est conséquence de R4, dont l'unique prémisse E appartient à la base.

Enfin, le fait C étant démontré (et donc ajouté à la base), le fait D l'est aussi, puisqu'il est une conséquence de la règle R6 dont C est une prémisse. Finalement, le but G a été atteint par le chemin :

$$(G) \leftarrow (F) \leftarrow \begin{cases}(A)\\(E)\end{cases} \leftarrow \begin{cases}(C)\\(D) \leftarrow (C) \leftarrow (E)\\(F)\end{cases}$$

Moteur d'inférence d'ordre 1

L'exemple qui nous a servi de support pour montrer la différence entre chaînage avant et chaînage arrière utilise un moteur d'inférence d'ordre zéro (on parle de logique d'ordre zéro). Cela signifie que les prémisses et les conséquences des règles sont des faits explicites. Par contre l'exemple portant sur les équations algébriques faisait appel à un moteur d'inférence d'ordre un : les prémisses et les conséquences sont des variables et peuvent s'appliquer à toute une classe de faits, ainsi la règle : $X+A=0 \Rightarrow X=-A$ est vraie pour $A=0,1,2$ etc., A peut même être une formule algébrique. Cette seconde catégorie de moteurs d'inférence possède une capacité de traitement bien supérieure car elle permet un niveau d'abstraction supplémentaire. Pour cette raison, les systèmes experts à moteurs d'inférence d'ordre zéro sont cantonnés à des domaines de type "initiation" ou ludiques.

D'autres structures de systèmes experts

La structure de systèmes experts que nous avons présentée est la plus classique, mais il en existe bien d'autres. Ainsi le programme BALANCE, que nous vous soumettons plus loin, tente par un moyen original dont nous allons vous faire la description, de reproduire le raisonnement qui conduit un médecin à fournir un diagnostic.

PRÉSENTATION DU PROGRAMME BALANCE (DIAGNOSTIC)

COMMENT L'UTILISER ?

La petite base de connaissance associée à cette application est du domaine de la médecine. Elle constitue un exemple à ne pas prendre au sérieux et ce chapitre contient tout les renseignements nécessaires pour constituer une base de connaissance dans un domaine différent (diagnostic de panne en automobile ou en informatique par exemple).

Ce logiciel connaît deux commandes :

– la commande L qui provoque l'affichage de la liste des maladies connues par le logiciel ;
– la commande D, comme diagnostic, qui, comme son nom l'indique délivre un diagnostic au terme d'un dialogue constitué de questions (de la part de l'ordinateur) et de réponses (de votre part) sous forme d'une lettre, O pour oui, N pour non et P pour peut-être (à n'utiliser que lorsque vous ignorez la réponse). A chaque stade du dialogue, la maladie la plus probable est affichée dans la fenêtre droite, accompagnée de la liste de ses symptômes. Lorsqu'une maladie dépasse un

DIAGNOSTIC

ROUGEOLE : 100 %

DIALOGUE

? o
ES-CE QUE LE FAIT SUIVANT EST VRAI:
toux
? o
ES-CE QUE LE FAIT SUIVANT EST VRAI:
les yeux sont rouges et larmoyants
? o
ES-CE QUE LE FAIT SUIVANT EST VRAI:
ecoulement nasal et eternuements
? o
ES-CE QUE LE FAIT SUIVANT EST VRAI:
nous sommes en automne ou en hiver
? n

O=oui N=non P=peut-etre F=fin

MALADIES/SYMPTOMES

**** ROUGEOLE ****
- 1 -
le patient n'a jamais eu la rougeole
- 2 -
eruption sans demangeaisons par macules confluentes (taches qui se reunissent), descendantes du haut en bas du corps
- 3 -
fievre plusieurs jours avant l'eruption
- 4 - toux
- 5 -
les yeux sont rouges et larmoyants
- 6 -
ecoulement nasal et eternuements

COMMANDES : L= liste maladies/symptomes D = diagnostic

pourcentage de trente, celle-ci est affichée, accompagnée de son pourcentage de vraisemblance (voir partie "comment ça marche" pour en savoir plus sur la signification de ce pourcentage). Si plusieurs maladies dépassent trente, elles sont classées et affichées dans cette même fenêtre, à concurrence de trois.

COMMENT ÇA MARCHE ?

Il faut aller chercher le principe du fonctionnement de ce logiciel dans la structure de la base de connaissance (c'est-à-dire la base qui contient les informations relatives aux maladies). Celle-ci se décompose en trois parties :

– la liste des symptômes, toutes maladies confondues ;
– la liste des interactions entre symptômes ;
– la liste des maladies, à chacune d'elle étant associée une liste de symptômes affectée de coefficients.

LA LISTE DES SYMPTOMES

Celle-ci est composée :

– d'un numéro, qui peut être quelconque, et dont la présence évite une refonte totale de la base lorsque l'on désire insérer un nouveau symptôme ;
– d'une formulation abrégée, destinée à construire rapidement la liste des symptômes par maladie ;
– d'une formulation in extenso, qui servira à interroger clairement l'utilisateur.

LA LISTE DES INTÉRACTIONS ENTRE SYMPTOMES

Ici réduite à sa plus simple expression, elle peut prendre une importance plus grande pour des bases de connaissance de nature différente. Son rôle est d'éviter les questions idiotes (lors de la recherche du diagnostic), dans le genre "est-ce que la fièvre est comprise entre 38 et 39" alors que précédemment, l'utilisateur a répondu "oui" à la question "est-ce que la fièvre est supérieure à 39". La formulation de chaque itéraction est du type :

IF liste de symptômes (vrai ou faux),
THEN liste de symptômes (vrai ou faux).

LA LISTE DES MALADIES

Chaque maladie (désignée par son nom), est suivie de ses symptômes accompagné de coefficients dans le cas où ils sont vérifiés et dans le cas contraire.

LA RECHERCHE D'UN DIAGNOSTIC

Lors d'une tentative de diagnostic, chaque maladie possède un score qui est fonction des réponses de l'utilisateur aux questions du logiciel. Ces questions sont du type : "est-ce que tel symptôme (affiché in extenso) est vérifié".

Le choix d'un symptôme est déterminé suivant deux règles :

– le symptôme doit être un de ceux de la maladie qui a obtenu jusqu'alors le meilleur score (donc qui est provisoirement retenue) ;

– la réponse à la question n'est pas encore connue (que ce soit par interrogation directe ou par intéraction entre symptômes).

```
DIAGNOSTIC
GRIPPE : 100 %

DIALOGUE
ES-CE QUE LE FAIT SUIVANT EST VRAI:
douleurs diffuses (articulations-mus
cles-tete)
? o
ES-CE QUE LE FAIT SUIVANT EST VRAI:
le patient est abattu
? o
ES-CE QUE LE FAIT SUIVANT EST VRAI:
frissons
? o
ES-CE QUE LE FAIT SUIVANT EST VRAI:
troubles respiratoires et digestifs
? o
O=oui N=non P=peut-etre F=fin

MALADIES/SYMPTOMES
**** GRIPPE ****
 - 1 -
la fievre est superieure a 39 degres

 - 2 - le debut a ete brutal
 - 3 -
douleurs diffuses (articulations-mus
cles-tete)
 - 4 - le patient est abattu
 - 5 - frissons
 - 6 -
troubles respiratoires et digestifs

COMMANDES :   L= liste maladies/symptomes   D = diagnostic
```

Après chaque question, les trois maladies ayant obtenu le meilleur score sont classées et affichées, à la condition toutefois d'avoir obtenu un score supérieur à 30 % (le score de 100 % correspondant à la vérification de tous les symptômes). Ce classement n'est qu'indicatif et doit être sujet à interprétation. Un score de 60 % pour la scarlatine ne signifie pas que le patient a 60 % de la scarlatine, mais qu'il vérifie 60 % de ses symptômes, étant entendu que la médecine n'est pas une science exacte, et qu'il n'est pas toujours possible de mettre en évidence tous les symptômes d'une maladie.

D'autre part il faut savoir que la scarlatine est une forme d'angine, donc si le patient contracte la scarlatine, l'angine sera certainement classée en second.

```
DIAGNOSTIC                               MALADIES/SYMPTOMES
OTITE : 83 %                             **** OTITE ****
                                          - 1 - douleur dans l'oreille
                                          - 2 - baisse de l'audition
                                          - 3 - fievre
                                          - 4 - vomissements
DIALOGUE
ES-CE QUE LE FAIT SUIVANT EST VRAI:
fievre
? o
ES-CE QUE LE FAIT SUIVANT EST VRAI:
douleur a la deglutition
? n
ES-CE QUE LE FAIT SUIVANT EST VRAI:
douleur dans l'oreille
? o
ES-CE QUE LE FAIT SUIVANT EST VRAI:
baisse de l'audition
? o

O=oui N=non P=peut-etre F=fin
COMMANDES :   L= liste maladies/symptomes   D = diagnostic
```

PRINCIPALES VARIABLES

Symptômes(i,j)	symptômes j des maladies i
maladie$(i)	nom de la maladie i
nsympt(i)	nombre de symptômes de la maladie i
symptôme(i)	numéro du symptôme i. Contrairement aux apparences, i et symptôme(i) ne font pas double emploi, ce dernier permettant l'insertion de nouveaux symptômes, sans contrainte de localisation dans le listing.
sympt.abrege$(i)	nom abrégé du symptôme i
symptôme$(i)	nom in extenso du symptôme i
valeur(i,j,k)	valeur dont est incrémenté le score de la maladie i si le j ième symptôme de celle-ci est vérifiée (k=0) ou non vérifié (k=1).
note.globale(i)	score atteint par la maladie i au cas où tous ses symptômes sont vérifiés.
score(i)	score atteint par la maladie i durant la recherche du diagnostic.
mliste(i)	liste des maladies candidates classées suivant leur score
mscore(i)	liste de ces scores, classés.

```
10 REM********************
20 REM  BALANCE  (T. L.-A)
30 REM********************
40 MODE 2
50 GOSUB 3030
60 REM===============
70 REM initialisation
80 REM===============
90 DEFINT a-z
100 nmaladies=20
110 nsymptomes.par.maladie=12
120 nsymptomes=200
130 vrai.ou.faux=3
140 inconnu=0:faux=1:vrai=2:peut.etre=3
150 non=0:oui=1
160 DIM symptomes(nmaladies,nsymptomes.par.maladie)
170 DIM maladie$(nmaladies)
180 DIM nsympt(nmaladies)
190 DIM symptome(nsymptomes)
200 DIM sympt.abrege$(nsymptomes)
210 DIM symptome$(nsymptomes)
220 DIM valeur(nmaladies,nsymptomes.par.maladie,vrai.ou.faux)
230 DIM note.globale(nmaladies)
240 DIM score(nmaladies)
250 DIM mliste(nmaladies)
260 DIM mscore(nmaladies)
270 REM================================================
280 REM symptomes : numero, nom abrege, nom in extenso
290 REM les numeros ne sont pas obliges de se suivre
300 REM ce qui facilite l'insertion
310 REM================================================
320 DATA 1,t>39, la fievre est superieure a 39 degres
330 DATA 2,38<t<39,la fievre est comprise entre 38 et 39 degres
340 DATA 4,t=normale, la temperature est normale
350 DATA 5,fievre,fievre
360 DATA 6,abattu, le patient est abattu
370 DATA 7,debut brutal,le debut a ete brutal
380 DATA 8,douleur deglu,douleur a la deglutition
390 DATA 9,gorge enflammee, la gorge est enflammee
400 DATA 10,ganglions.c, ganglions cervicaux
410 DATA 11,douleur oreille,douleur dans l'oreille
420 DATA 12,baisse audition,baisse de l'audition
430 DATA 13,courbatures,courbatures
440 DATA 14,frissons,frissons
450 DATA 15,malaise gen.,malaise general
460 DATA 16,douleurs diffuses, douleurs diffuses (articulations-muscles-tete)
470 DATA 17,troubles resp. et dig.,troubles respiratoires et digestifs
480 DATA 18,jamais eu scarlatine,le patient n'a jamais eu la scarlatine
```

```
490 DATA 19,eruption en plaques,eruption en plaques etendues sur le tronc et la
racine des membres
500 DATA 20,etat particulier de la langue,etat de la langue --> pourtour rouge f
raise et centre recouvert d'un enduit blanchatre
510 DATA 21,jamais eu rougeole,le patient n'a jamais eu la rougeole
520 DATA 22,eruption b,"eruption sans demangeaisons par macules confluentes (tac
hes qui se reunissent), descendantes du haut en bas du corps"
530 DATA 23,fievre precoce,fievre plusieurs jours avant l'eruption
540 DATA 24,toux,toux
550 DATA 25,yeux rouges/larmoyants,les yeux sont rouges et larmoyants
560 DATA 26,nez,ecoulement nasal et eternuements
570 DATA 27,jamais eu oreillons,le patient n'a jamais eu les oreillons
580 DATA 28,tumefaction part.,tumefaction douloureuse des deux cotes de la tete
en avant des oreilles
590 DATA 29,douleur oreille,douleur dans l'oreille (qui provoque chez le nourris
son cris et insomnies)
600 DATA 30,vomissements,vomissements
610 DATA 31,b.audition,baisse de l'audition
620 DATA 32,homme,le patient est un homme
630 DATA 33,brulures miction,brulures intenses a la miction (en urinant)
640 DATA 34,ecoul.verge,ecoulement purulent de la verge
650 DATA 35,automne/hiver,nous sommes en automne ou en hiver
660 DATA 36,obstruction nasale,obstruction nasale
670 DATA 37,diarrhee,diarrhee
680 DATA 38,nausees,le malade ressent des nausees
690 DATA 39,coliques,coliques
700 DATA fin
710 REM======= remplissage de symptome (i) =========
720 READ i$
730 IF i$="fin" THEN nsympt=i:GOTO 1750
740 i=VAL(i$)
750 READ s$:sympt.abrege$(i)=s$
760 READ s$:symptome$(i)=s$
770 GOTO 720
780 REM===============================
790 REM interaction entre les symptomes
800 REM===============================
810 IF symptome(1)=vrai THEN symptome(2)=faux:symptome(4)=faux:symptome(5)=vrai
820 IF symptome(2)=vrai THEN symptome(1)=faux:symptome(4)=faux:symptome(5)=vrai
830 IF symptome(4)=vrai THEN symptome(1)=faux:symptome(2)=faux:symptome(5)=faux
840 IF symptome(5)=faux THEN symptome(1)=faux:symptome(2)=faux:symptome(4)=vrai
850 RETURN
860 REM=========================================
870 REM maladie, leurs symptomes, suivis de leur
880 REM ponderation :
890 REM - s'ils sont verifies
900 REM - s'ils ne sont pas verifies
910 REM=========================================
920 DATA ANGINE
```

```
930 DATA fievre,10,-10
940 DATA douleur deglu,10,-1000
950 DATA gorge enflammee,10,-10
960 DATA ganglions.c,8,-8
970 DATA courbatures,7,0
980 DATA malaise gen.,5,-5
990 DATA frissons,4,0
1000 DATA fin
1010 REM------
1020 DATA GRIPPE
1030 DATA t>39,10,-1000
1040 DATA debut brutal,10,-2
1050 DATA douleurs diffuses,10,-1000
1060 DATA abattu,10,0
1070 DATA frissons,6,0
1080 DATA troubles resp. et dig.,5,0
1090 DATA fin
1100 REM------
1110 DATA SCARLATINE
1120 DATA jamais eu scarlatine,0,-1000
1130 DATA fievre,10,-1000
1140 DATA douleur deglu,5,-1000
1150 DATA gorge enflammee,5,-1000
1160 DATA ganglions.c,4,-4
1170 DATA courbatures,3,0
1180 DATA frissons,2,0
1190 DATA malaise gen.,5,-5
1200 DATA eruption en plaques,20,-1000
1210 DATA etat particulier de la langue,20,-1000
1220 DATA fin
1230 REM------
1240 DATA ROUGEOLE
1250 DATA jamais eu rougeole,0,-1000
1260 DATA eruption b,10,-1000
1270 DATA fievre precoce,10,-1000
1280 DATA toux,10,-1000
1290 DATA yeux rouges/larmoyants,10,-1000
1300 DATA nez,8,0
1310 DATA fin
1320 REM------
1330 DATA OREILLONS
1340 DATA jamais eu oreillons,0,-1000
1350 DATA 38<t<39,10,-2
1360 DATA malaise gen.,5,0
1370 DATA tumefaction part.,25,-1000
1380 DATA fin
1390 REM------
1400 DATA OTITE
1410 DATA douleur oreille,10,-1000
```

```
1420 DATA b.audition,10,-1000
1430 DATA fievre,5,-2
1440 DATA vomissements,5,0
1450 DATA fin
1460 REM------
1470 DATA BLENNHORRAGIE
1480 DATA homme,0,-1000
1490 DATA brulures miction,10,-1000
1500 DATA ecoul.verge,10,-1000
1510 DATA fievre,3,0
1520 DATA fin
1530 REM------
1540 DATA CORYZA SAISONNIER (RHUME)
1550 DATA automne/hiver,0,-5
1560 DATA t=normale,1,-1000
1570 DATA nez,10,-1000
1580 DATA obstruction nasale,10,0
1590 DATA fin
1600 REM------
1610 DATA INFECTION INTESTINALE
1620 DATA fievre,3,0
1630 DATA diarrhee,10,-5
1640 DATA coliques,10,-1
1650 DATA vomissements,5,0
1660 DATA nausees,5,0
1670 DATA fin
1680 REM------
1690 REM===================
1700 DATA fin des maladies
1710 REM=========================================
1720 REM remplissages des variables representant
1730 REM les caracteristiques de chaque maladie
1740 REM=========================================
1750 READ maladie$
1760 IF maladie$="fin des maladies" THEN nmaladies=maladie-1:GOTO 1960
1770 maladie$(maladie)=maladie$
1780 READ symptome$
1790 IF symptome$="fin" THEN nsympt(maladie)=j-1:maladie=maladie+1:j=0:GOTO 1750

1800 REM--- recherche du numero de symptome$ ---
1810   i=0
1820   IF symptome$=sympt.abrege$(i) THEN 1860
1830   i=i+1
1840   IF i>nsympt THEN PRINT "ERREUR, symptome '";symptome$;"' inconnu":END
1850   GOTO 1820
1860 symptomes(maladie,j)=i
1870 READ v
1880 valeur(maladie,j,vrai)=v
1890 note.globale(maladie)=note.globale(maladie)+v
```

```
1900 READ v
1910 valeur(maladie,j,faux)=v
1920 j=j+1:GOTO 1780
1930 REM=====================
1940 REM entree d'une commande
1950 REM=====================
1960 INPUT "COMMANDE";c$
1970 c$=UPPER$(c$)
1980 IF c$="D" THEN 2050
1990 IF c$="L" THEN 2600
2000 PRINT "COMMANDE INCONNUE"
2010 GOTO 1960
2020 REM==============================
2030 REM entree des donnees - dialogue
2040 REM==============================
2050 GOSUB 2350
2060 CLS:m=0:ml=1000
2070 maladie=0
2080 j=0
2090 m=mliste(maladie)
2100 IF mliste(0)<>ml THEN ml=mliste(0):GOSUB 2690
2110 s=symptomes(m,j)
2120 IF symptome(s)<>inconnu OR score(m)<-35 THEN 2280
2130 PRINT "ES-CE QUE LE FAIT SUIVANT EST VRAI:"
2140 PRINT symptome$(s)
2150 INPUT rep$
2160 rep$=UPPER$(rep$)
2170 IF rep$="O" THEN rep=vrai:GOTO 2220
2180 IF rep$="N" THEN rep=faux:GOTO 2220
2190 IF rep$="P" THEN rep=peut.etre:GOTO 2220
2200 IF rep$="F" THEN 2910
2210 PRINT "REPONDRE PAR O,N,P ou F":GOTO 2150
2220 symptome(s)=rep
2230 GOSUB 810:REM interactions entre symptomes
2240 GOSUB 2780:REM calcul des scores des maladies
2250 GOSUB 2350:REM classement et diagnostic provisoire
2260 GOTO 2070
2270 REM---------------
2280 j=j+1
2290 IF j>nsympt(m) THEN maladie=maladie+1:IF maladie=nmaladies THEN 2900 ELSE 2
080
2300 GOTO 2110
2310 REM===============================================
2320 REM classement et affichage des maladies candidates
2330 REM===============================================
2340 REM--- transfert des scores en vue du classement ---
2350 FOR i=0 TO nmaladies
2360   mliste(i)=i
2370   mscore(i)=score(i)
```

```
2380 NEXT i
2390 REM--- tri des maladies suivant leur score ---
2400 FOR i=nmaladies TO 1 STEP-1
2410   s=0
2420   FOR j=1 TO i
2430     IF mscore(j)<=mscore(j-1) THEN 2460
2440     t=mscore(j):mscore(j)=mscore(j-1):mscore(j-1)=t:s=j
2450     tm=mliste(j):mliste(j)=mliste(j-1):mliste(j-1)=tm
2460   NEXT j
2470   i=s
2480 NEXT i
2490 REM--- affichage ---
2500 CLS#1
2510 FOR maladie=0 TO nmaladies
2520   m=mliste(maladie)
2530   IF score(m)<25 THEN 2550
2540   PRINT#1,maladie$(m);" :";score(m);"%"
2550 NEXT maladie
2560 RETURN
2570 REM========================
2580 REM liste maladies/symptomes
2590 REM========================
2600 CLS
2610 FOR ml=0 TO nmaladies
2620   GOSUB 2690
2630   INPUT "APPUYEZ SUR ENTER",rep$
2640   CLS
2650 NEXT ml
2660 CLS#2
2670 GOTO 1960
2680 REM==== liste de symptomes d'une maladie ====
2690 CLS#2
2700 PRINT#2,"**** ";maladie$(ml);" ****"
2710 FOR symptome=0 TO nsympt(ml)
2720   PRINT#2," -"symptome+1;"- ";symptome$(symptomes(ml,symptome))
2730 NEXT symptome
2740 RETURN
2750 REM==============================
2760 REM calcul des scores des maladies
2770 REM==============================
2780 FOR maladie=0 TO nmaladies
2790   score(maladie)=0
2800   FOR i=0 TO nsympt(maladie)
2810     etat=symptome(symptomes(maladie,i))
2820     score(maladie)=score(maladie)+valeur(maladie,i,etat)
2830   NEXT i
2840   score(maladie)=100*score(maladie)/note.globale(maladie)
2850 NEXT maladie
2860 RETURN
```

```
2870 REM=======================================
2880 REM fin de diagnostic - re-initialisation
2890 REM=======================================
2900 PRINT "JE NE PEUX AFFINER DAVANTAGE":PRINT "LE DIAGNOSTIC"
2910 FOR maladie=0 TO nmaladies
2920   score(maladie)=0
2930 NEXT maladie
2940 FOR s=0 TO nsympt
2950   symptome(s)=0
2960 NEXT s
2970 PRINT "POUR REVENIR AU MODE COMMANDE":INPUT "APPUYEZ SUR ENTER",rep$
2980 CLS:CLS#1:CLS#2
2990 GOTO 1960
3000 REM=======================
3010 REM presentation de l'ecran
3020 REM=======================
3030 SYMBOL 255,255,0,255,0,255,0,255,0
3040 MODE 2
3050 FOR y=7 TO 23 STEP 16
3060   LOCATE 2,y:PRINT STRING$(39,255);
3070 NEXT y
3080 LOCATE 2,1:PRINT STRING$(39,255);
3090 LOCATE 41,1:PRINT STRING$(39,255);
3100 FOR y=1 TO 23
3110   LOCATE 1,y:PRINT CHR$(143);
3120   LOCATE 41,y:PRINT CHR$(143);
3130   LOCATE 80,y:PRINT CHR$(143);
3140 NEXT y
3150 LOCATE 16,7:PRINT " DIALOGUE ";
3160 LOCATE 15,1:PRINT " DIAGNOSTIC ";
3170 LOCATE 7,23:PRINT "O=oui N=non P=peut-etre F=fin";
3180 LOCATE 52,1:PRINT " MALADIES/SYMPTOMES ";
3190 LOCATE 1,y:PRINT STRING$(80,207);
3200 LOCATE 1,25:PRINT STRING$(80,143);
3210 LOCATE 5,25:PRINT " COMMANDES : ";
3220 LOCATE 25,25:PRINT " L= liste maladies/symptomes ";
3230 LOCATE 61,25:PRINT " D = diagnostic ";
3240 WINDOW 3,38,9,21
3250 WINDOW#1,3,38,3,5
3260 WINDOW#2,43,78,3,22
3270 MOVE 12,380
3280 DRAW 316,380:DRAW 316,309:DRAW 12,309:DRAW 12,380
3290 MOVE 332,380
3300 DRAW 628,380:DRAW 628,37:DRAW 332,37:DRAW 332,380
3310 MOVE 12,284
3320 DRAW 316,284:DRAW 316,53:DRAW 12,53:DRAW 12,284
3330 RETURN
```

TROISIÈME PARTIE : ACTION

CHAPITRE 5
LES ROBOTS

MISE AU POINT

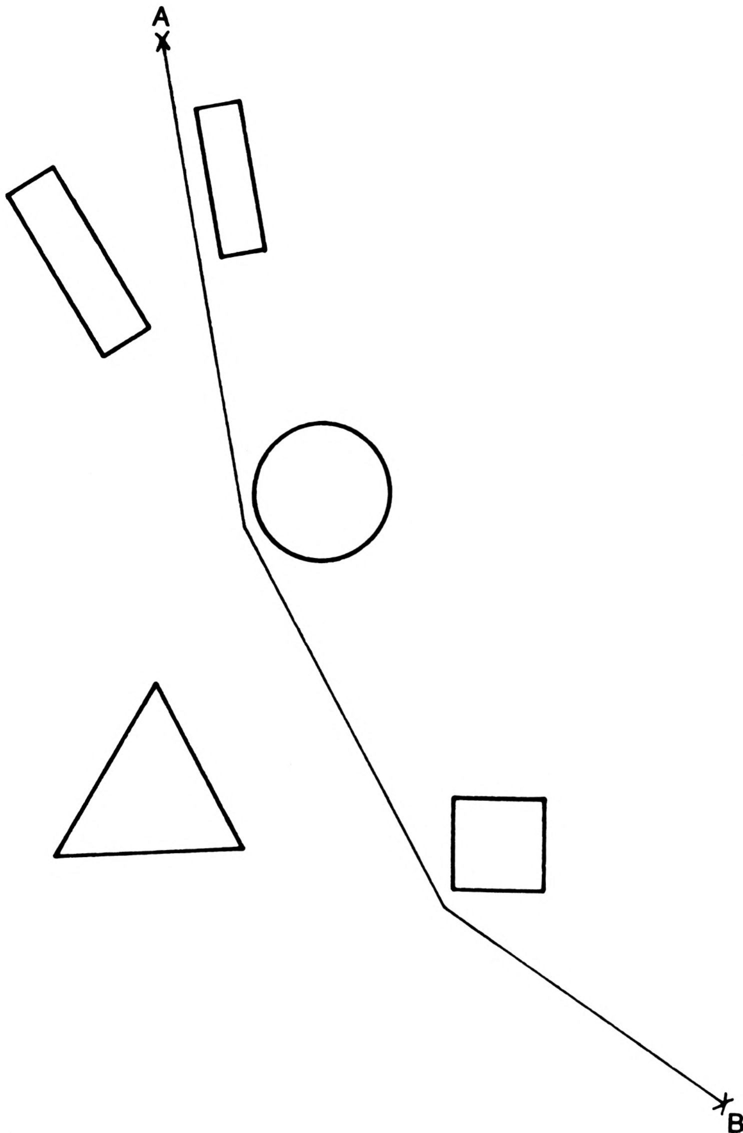
Nous ne prétendons pas ici traiter un sujet comme la robotique de manière exhaustive quand l'Amstrad, sur lequel nous présenterons quelques applications propres à ce domaine, ne dispose d'aucun organe mécanique. Tout du moins nous ne comptons pas aborder tous les aspects de ce vaste sujet, en particulier certains aspects dont nous avons déjà parlé (vision, décision) ou d'autres qui nécessiteraient pour être mis en œuvre des moyens dont vous ne disposez pas pour la plupart. La question alors posée est la suivante : "Que peut-on faire en robotique avec un micro-ordinateur comme l'Amstrad ?". La réponse vous étonnera sans doute : "presque tout" ! Rappelons que parmi les trois phrases de toute initiative intelligente (PERCEPTION, RÉFLEXION, ACTION), deux sont en contact étroit avec le monde extérieur : la perception et l'action, illustrées par la robotique.

Or, tout comme la reconnaissance de formes ou de la parole, qui interviennent dans les phases de perception, l'action (la robotique en est une application immédiate) implique une activité intellectuelle importante. Toute cette activité intellectuelle qui préfigure le mouvement et qui fait partie intégrante de l'action (sauf peut-être dans les RAMBO de Silverster Stallone) pourra être simulée sur un micro puisque qu'elle ne nécessite aucun organe mécanique. Comme il est toujours un peu frustrant de préparer une action sans la mettre effectivement en œuvre, nous simulerons celle-ci graphiquement sur l'écran.

L'ÉTENDUE DU PROBLÈME

Un exemple illustre bien notre propos : le problème de la préhension d'objets. Pour qu'un robot parvienne à se saisir d'un objet, il doit faire précéder son mouvement par plusieurs étapes purement intellectuelles. Il doit d'abord "comprendre" l'ordre que l'on veut lui faire exécuter. Cet ordre lui sera par exemple transmis en langage naturel. Il doit ensuite identifier l'objet, sa position, etc., puis calculer une trajectoire optimale pour l'atteindre en ajustant accélération et décélération, en évitant les obstacles, etc. Par exemple le robot VESA utilise plusieurs tactiques différentes pour déterminer sa trajectoire. La plus simple consiste à se diriger droit devant lui vers l'objectif. S'il rencontre un obstacle, il le contourne puis reprend sa marche initiale par une série de rotations.

Une stratégie plus élaborée suppose une connaissance a priori de l'environnement. Les obstacles sont, soit modélisés par des formes géométriques soit grossièrement décrits par un opérateur dans un langage particulier. VESA calcule alors AVANT d'entamer le mouvement, une trajectoire minimale tenant compte de ces obstacles.



Trajectoire optimale de A vers B

Toujours avant même d'amorcer le moindre mouvement, un robot estime la pression qu'il exercera sur l'objet à saisir. Pour cela, il enregistre l'aspect de l'objet et recherche dans ses bases de données toutes les informations relatives à ce type d'objet : résistance, consistance, etc. La même démarche intellectuelle est effectuée pour ce qui concerne le retour du robot après la préhension. A ceci près que la fragilité éventuelle de l'objet saisi complique encore un peu le travail intellectuel de notre robot !

Et même le mouvement entamé, le travail intellectuel continue. En effet des obstacles imprévus peuvent survenir qu'il lui faudra éviter (en le contournant ? en les escaladant ? en les détruisant ?). Ce n'est qu'au moment de la préhension effective de l'objet que le robot, grâce à des capteurs, se rend compte de sa résistance (élasticité, consistance), de sa température, de son poids. En fonction de ces nouvelles informations, il adapte la pression de ses "doigts", éventuellement reconsidère sa trajectoire de retour, etc.

Comme vous le voyez, le travail d'un robot n'est pas simple. Et une grande part de ce travail peut être effectuée sur un micro-ordinateur. Ceux-ci se prêtent très bien à ce genre d'applications et il n'est pas rare de piloter un engin mécanique à partir d'un micro tout à fait ordinaire.

L'ordinateur se charge alors de la part de raisonnement présente dans toute action, et la machine se contente de figurer les bras ou les jambes dans cette association de troisième type. Parmi les différents programmes qui illustrent ce chapitre sur la dernière phase du processus d'intelligence, l'action, citons ROBOT, capable de manipuler des objets en obéissant à des ordres donnés en langage naturel. En le voyant réagir à vos commandes et tenir compte de son environnement (toute l'action est simulée sur l'écran) vous prendrez sans doute conscience de la complexité de la préparation de l'action... tout cela en vous amusant bien entendu ! De même SOPHIE, extraordinaire de réalisme, simule avec ses lèvres toutes les phrases que vous rentrerez au clavier. Certes, la part d'intelligence dans ce type d'action se limite à peu de choses mais elle existe néanmoins : SOPHIE possède la capacité d'identifier les phonèmes des mots que vous lui demanderez de prononcer, puis d'associer à chacun de ces phonèmes un certain mouvement de lèvres. Pas si évident qu'il n'y paraît !

LES APPLICATIONS DES ROBOTS

Peut-être encore davantage que les autres domaines de l'intelligence artificielle, celui-ci constitue de la graine de bouleversement social. Les robots constituent un enjeu économique sans précédent et les gouvernements l'ont bien compris incitant les centres de recherche à rechercher à tout va, les industries à s'équiper sans attendre. Les japonais ont fait des robots un axe prioritaire de leurs investigations vers la fameuse cinquième génération d'ordinateurs. Et quand les spécialistes parlent de robots, ils n'évoquent pas tant le côté "machinerie" qui évolue peu par rapport aux automates déjà existants et autres machines-outils, mais bien plutôt, nous n'insisterons jamais assez, l'aspect "intelligence" qu'il convient désormais de leur associer.

Grâce justement à cette intelligence, qui justifie d'ailleurs la présence de ce chapitre dans un ouvrage traitant de l'intelligence artificielle, les robots sont et seront de plus en plus, capables d'accomplir des tâches évoluées. Car l'ambition des industriels ne donne pas dans la demi-mesure : il s'agit tout simplement de remplacer des hommes, accomplissant des travaux certes répétitifs mais réclamant néanmoins une bonne capacité d'adaptabilité voire d'initiative, par des machines faites de métal et de silicium. C'est dire l'ampleur du mouvement qui s'amorce !

Parmi les nombreuses applications recencées à ce jour, quelques-unes marquent plus que d'autres par le côté "évolué" de la tâche qui incombe au robot.

- robot munis d'une caméra pour détecter les défauts d'une marchandise ;
- robot pour assembler des circuits électroniques sur des cartes ;
- robot pour trier des pièces automobiles ;

– robot pour peindre des voitures ;
– robot pour assembler des éléments mécaniques complexes ;
– robot capable de lire une partition et la jouer au piano ;
– robot soudeur, pour la manutention.

Le Japon dans le domaine des robots possède une avance confortable sur les autres pays. Parmi les plus grands constructeurs, on peut citer KAWASAKI, HITACHI, FUJITSU, MITSUBISHI, NISSAN. Remarquons en passant qu'il s'agit pour la plupart de constructeurs d'automobiles clairvoyants qui ont vu d'où venait le vent. Quant à la recherche, une action d'envergure entreprise dans ce pays mérite qu'on s'y arrête.

Le projet MUM (Methodology for Unmanned Machine-shop) a conduit à la réalisation d'un atelier flexible de production, c'est-à-dire entièrement automatisé !

LES FONCTIONS D'UN ROBOT

Si l'on récapitule en les formalisant les différentes fonctions remplies par un robot, on en relève cinq principales :

– la fonction PERCEPTION : le robot perçoit son environnement au moyen de capteurs et de senseurs ;

– la fonction COMMUNICATION : le robot communique avec un opérateur qui lui indique la marche à suivre, les ordres à exécuter, etc. ;

– la fonction DÉCISION : le robot utilise les résultats des deux précédentes fonctions pour choisir une trajectoire, un comportement, etc. ;

– la fonction ACTION : le robot met en œuvre sa décision pour réaliser le but fixé par un opérateur ;

– la fonction APPRENTISSAGE : le robot itère les quatre précédentes étapes et enregistre en mémoire tous les événements qui en découlent. Ils fourniront plus tard de la matière à la fonction décision.

LANGAGE NATUREL ET ROBOTS

Comme vous le constaterez dans vos investigations futures sur les trois logiciels qui illustrent ce chapitre, ROBOT (un de ces trois), possède une particularité assez rare et des plus intéressantes. Il obéit à des ordres indiqués en langage naturel. Communiquer avec les robots en langage naturel apparaît indiscutablement comme l'approche la plus logique et la plus inéluctable. En effet, de plus en plus, les robots sont amenés à se livrer à de véritables dialogues avec les opérateurs humains, tant la communication devient une de leurs fonctions prépondérantes. L'opérateur transmet non seulement des ordres, mais aussi des informations sur l'environnement, des requêtes (qui appellent des réponses), etc.

Dans un tel contexte, deux voies sont possibles : soit les opérateurs deviennent des spécialistes très pointus, soit ce sont les robots qui s'adaptent et comprennent le langage naturel, celui que parle n'importe qui (avec quelques restrictions tout de même). Comme il est toujours plus sûr d'adapter une machine (si c'est possible) qu'un homme, la solution consiste à appliquer aux robots les techniques les plus avancées de la reconnaissance de la parole, (cf. chapitre sur la reconnaissance de la parole).

QUELQUES EXEMPLES CÉLÈBRES

Sans faire l'inventaire de tous les robots passés, présents ou futurs, rendons néanmoins hommage à quelques "phénomènes" robotiques qui forcent l'admiration et incitent à la réflexion philosophique (jusqu'où iront-ils ??).

ELIZA simule l'activité d'un psychiatre, l'opérateur humain jouant le rôle du patient.

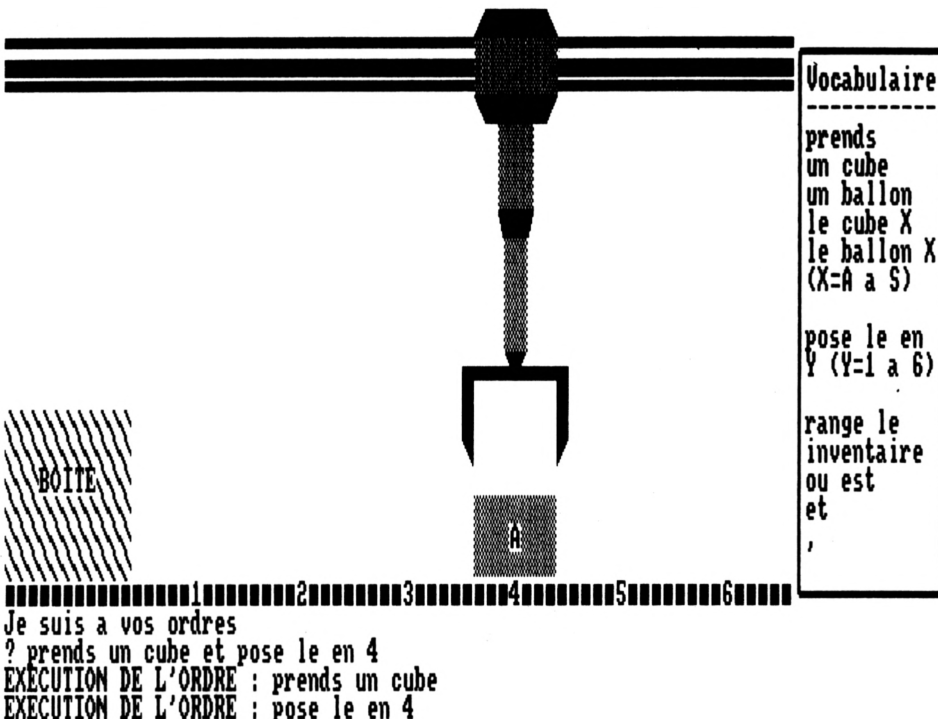
SHRDLU évolue dans un univers à trois dimensions contenant des boîtes, cylindres, etc. qu'il peut manipuler à l'aide d'un bras. L'opérateur humain communique avec SHRDLU en langage naturel.

Enfin un robot (CORDIER, 79) exécute des recettes de cuisines en mélangeant les ingrédients, calculant les temps de cuisson, etc.

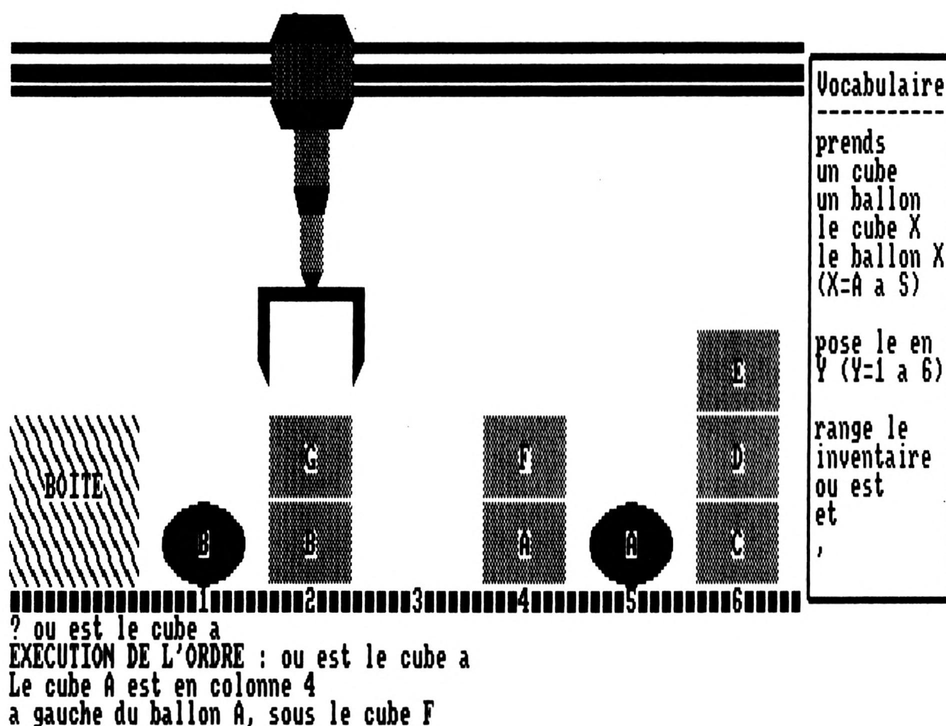
PRÉSENTATION DU PROGRAMME ROBOT (BRAS ROBOT PILOTABLE PAR PHRASES EN LANGAGE NATUREL)

COMMENT L'UTILISER ?

Vous pilotez, par des ordres tapés en langage naturel, un robot composé d'un bras se déplaçant sur une glissière et qui se termine par une pince. Il évolue dans une sorte de micro-monde et peut éventuellement aller chercher des cubes et des ballons dans une boîte.



Son vocabulaire est affiché dans une fenêtre, vous pouvez assembler ses mots comme bon vous semble, de manière à former des ordres ou des séries d'ordres, séparés par "et" ou une virgule. Chaque ordre est exécuté le cas échéant, sur l'écran, grâce à une superbe animation graphique.



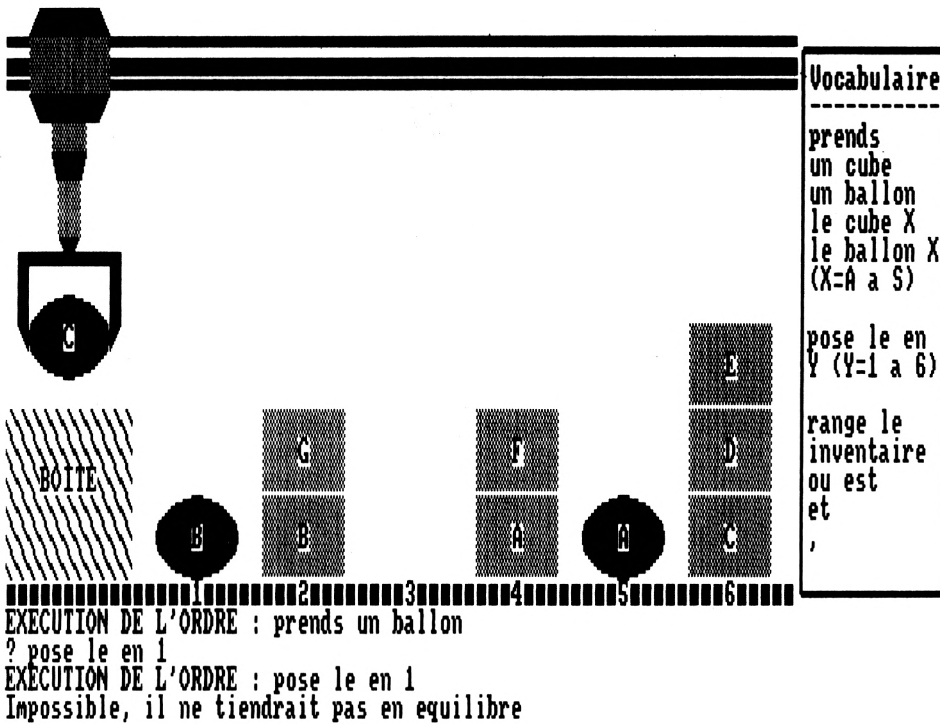
COMMENT ÇA MARCHE ?

Le fonctionnement de ce programme est régi par trois grands modules, parfaitement délimités dans le listing.

Le premier d'entre eux définit le **graphisme statique.** C'est-à-dire la préparation de l'écran (fenêtres, cadres, liste du vocabulaire), la définition des caractères graphiques, et enfin la création des chaînes de caractères graphiques qui forment les différents états du bras, ainsi que l'aspect des objets.

Le second module contient de nombreux sous-programmes chargés de **l'animation et de la modification de l'univers** représenté par le tableau UNIVERS(6,3).

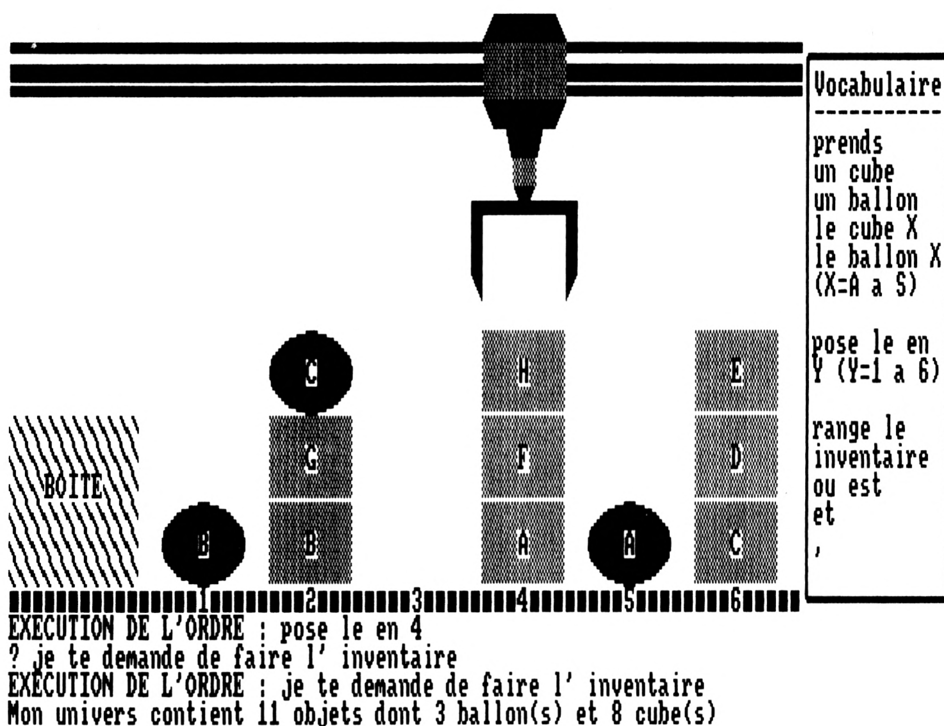
- déplacement de la pince d'un point à un autre ;
- saisie d'un cube ou d'un ballon dans la boîte ;
- rangement dans la boîte de l'objet que tient la pince ;
- dépose de l'objet que tient la pince à une certaine position ;
- dégagement d'un objet pour le rendre accessible à la pince.



Chacun de ces sous-programmes est précédé d'un en-tête spécifiant sa fonction et ses conditions d'entrée. Il est à noter qu'ils s'appellent les uns les autres, ainsi, le sous-programme qui dégage un objet utilise les programmes de saisie et de rangement, qui eux-mêmes appellent le programme de déplacement de la pince.

Le troisième module gère **le dialogue**, c'est-à-dire l'analyse syntaxique de phrases, la création des réponses, et l'exécution des ordres. Ce module procède par trois étapes :

– découpage de la phrase en ordres ;
– découpage de chaque ordre en mots ;
– comparaison de la série de mots avec l'arbre des ordres autorisés, cette opération néglige les mots inconnus ;
– exécution de l'ordre.



L'introduction de mots non connus du logiciel ne perturbe pas son fonctionnement.

PRINCIPALES VARIABLES

univers(i,j)	état case de coordonnées i et j de l'univers
xobjet(i)	abscisse de l'objet i dans l'univers
yobjet(i)	ordonnée de l'objet i dans l'univers
forme.objet(i)	forme de l'objet i (carré ou rond)
lettre.objet(i)	lettre de l'objet i (A à Z)

```
10 REM******************
20 REM   ROBOT  T. L.-A.
30 REM******************
40 REM******************************
50 REM 1 ere partie : initialisation
60 REM******************************
70 MODE 2
80 vide=0:ballon=1:cube=2:oui=1:non=0
90 DIM bras$(10),dir$(3),forme$(2),mot$(255)
100 DIM univers(6,3),xobjet(26),yobjet(26)
110 DIM forme.objet(26),lettre.objet$(26),existe(26)
120 dir$(0)="sur le":dir$(1)="a gauche du"
130 dir$(2)="sous le":dir$(3)="a droite du"
```

```
140 forme$(1)="ballon":forme$(2)="cube"
150 REM--- caracteres graphiques ---
160 SYMBOL AFTER 235
170 SYMBOL 235,170,85,170,85,170,85,170,0
180 SYMBOL 236,7,3,1,0,255,255,255,255
190 SYMBOL 237,224,192,128,0,255,255,255,255
200 SYMBOL 238,0,0,0,0,0,192,240,252
210 SYMBOL 239,252,240,192,0,0,0,0,0
220 SYMBOL 240,255,255,255,0,0,0,255,255
230 SYMBOL 241,255,255,255,0,255,255,255
240 SYMBOL 242,255,255,127,127,63,63,31,31
250 SYMBOL 243,255,255,254,254,252,252,248,248
260 SYMBOL 244,0,0,0,0,0,3,15,63
270 SYMBOL 245,0,0,7,63,255,255,255,255
280 SYMBOL 246,0,63,255,255,255,255,255,255
290 SYMBOL 247,0,252,255,255,255,255,255,255
300 SYMBOL 248,0,0,224,252,255,255,255,255
310 SYMBOL 249,127,127,255,255,255,255,127,127
320 SYMBOL 250,254,254,255,255,255,255,254,254
330 SYMBOL 251,63,15,3,0,0,0,0,0
340 SYMBOL 252,255,255,255,255,63,7,0,0
350 SYMBOL 253,255,255,255,255,255,255,63,0
360 SYMBOL 254,255,255,255,255,255,255,252,0
370 SYMBOL 255,255,255,255,255,252,224,0,0
380 REM--- forme du ballon ---
390 DATA 244,245,246,143,247,248,238
400 DATA 249,143,143,143,143,250
410 DATA 32,251,252,253,143,254,255,239,32
420 FOR i=1 TO 7:READ a:b1$=b1$+CHR$(a):NEXT i
430 FOR i=1 TO 3:READ a:b21$=b21$+CHR$(a):NEXT i
440 FOR i=1 TO 3:READ a:b22$=b22$+CHR$(a):NEXT i
450 FOR i=1 TO 9:READ a:b3$=b3$+CHR$(a):NEXT i
460 REM--- forme du cube ---
470 FOR i=1 TO 7:c1$=c1$+CHR$(207):NEXT i
480 FOR i=1 TO 3:c2$=c2$+CHR$(207):NEXT i
490 FOR i=1 TO 7:c3$=c3$+CHR$(235):NEXT i:c3$=" "+c3$+" "
500 REM--- forme du coude ---
510 DATA 32,214,143,143,143,143,143,215,32,10,8,8,8,8,8,8,8,8
520 DATA 240,207,207,207,207,207,207,207,240,10,8,8,8,8,8,8,8,8
530 DATA 241,207,207,207,207,207,207,207,241,10,8,8,8,8,8,8,8,8
540 DATA 32,213,143,143,143,143,143,212,32,10,8,8,8,8,8,8
550 FOR i=1 TO 73:READ a:coude$=coude$+CHR$(a):NEXT i
560 REM--- forme d'un element de bras ---
570 DATA 207,207,207,10,8,8,8
580 FOR i=1 TO 7:READ a:e1$=e1$+CHR$(a):NEXT i
590 DATA 242,143,243,10,8,8,8
600 FOR i=1 TO 7:READ a:e2$=e2$+CHR$(a):NEXT i
610 DATA 217,207,219,10,8,8,8
620 FOR i=1 TO 7:READ a:e3$=e3$+CHR$(a):NEXT i
```

```
630 REM--- forme de la pince ---
640 DATA 8,8,8,32,32,32,217,207,219,32,32,32,10,8,8,8,8,8,8,8,8,8,8
650 DATA 32,140,140,140,236,143,237,140,140,140,32,10,8,8,8,8,8,8,8,8,8,8,8
660 DATA 32,143,32,32,32,32,32,32,32,143,32,10,8,8,8,8,8,8,8,8,8,8,8
670 DATA 32,143,143,32,10,8,8,8,8,8,8,8,8,8,8,32,213
680 DATA 212,32,10,8,8,8,8,8,8,8,8,8,8,8,32
690 DATA 10,8,8,8,8,8,8,8,8
700 FOR i=1 TO 71:READ a:p1$=p1$+CHR$(a):NEXT i
710 FOR i=1 TO 16:READ a:p2$=p2$+CHR$(a):NEXT i
720 FOR i=1 TO 15:READ a:p3$=p3$+CHR$(a):NEXT i
730 FOR i=1 TO 9:READ a:p4$=p4$+CHR$(a):NEXT i
740 esp7$=STRING$(7,32):esp9$=STRING$(9,32)
750 o1$=esp7$:o2$=esp7$:o3$=esp9$:o4$="":GOSUB 2380
760 REM--- forme des bras des differentes longueurs ---
770 bras$(1)=e2$
780 bras$(2)=e1$+e2$
790 bras$(3)=e1$+e2$+e3$
800 bras$(4)=e1$+e1$+e2$+e3$
810 bras$(5)=e1$+e1$+e2$+e3$+e3$
820 bras$(6)=e1$+e1$+e1$+e2$+e3$+e3$
830 bras$(7)=e1$+e1$+e1$+e2$+e3$+e3$+e3$
840 bras$(8)=e1$+e1$+e1$+e1$+e2$+e3$+e3$+e3$
850 bras$(9)=e1$+e1$+e1$+e1$+e2$+e3$+e3$+e3$+e3$
860 bras$(10)=e1$+e1$+e1$+e1$+e1$+e2$+e3$+e3$+e3$+e3$
870 REM--- initialisation de l'ecran ---
880 WINDOW 1,80,22,25
890 WINDOW#1,69,79,3,20
900 LOCATE#2,1,2:PRINT#2,STRING$(67,240);
910 LOCATE#2,1,3:PRINT#2,STRING$(67,241);
920 LOCATE#2,1,21:PRINT#2,STRING$(67,233);
930 LOCATE#2,69,21:PRINT#2,STRING$(11,154);
940 LOCATE#2,69,2:PRINT#2,STRING$(11,154);
950 FOR x=1 TO 6:LOCATE#2,17+(x-1)*9,21:PRINT#2,RIGHT$(STR$(x),1);:NEXT x
960 FOR y=3 TO 20
970   LOCATE#2,68,y:PRINT#2,CHR$(149);
980   LOCATE#2,80,y:PRINT#2,CHR$(149);
990 NEXT y
1000 LOCATE#2,68,2:PRINT#2,CHR$(150);:LOCATE#2,80,2:PRINT#2,CHR$(156);
1010 LOCATE#2,68,21:PRINT#2,CHR$(147);:LOCATE#2,80,21:PRINT#2,CHR$(153);
1020 FOR y=15 TO 20
1030   LOCATE#2,1,y:PRINT#2,STRING$(11,201);
1040 NEXT y
1050 LOCATE#2,4,17:PRINT#2,"BOITE";
1060 PRINT#1,"Vocabulaire";STRING$(11,"-");
1070 PRINT#1,"prends"
1080 PRINT#1,"un cube":PRINT#1,"un ballon"
1090 PRINT#1,"le cube X":PRINT#1,"le ballon X";
1100 PRINT#1,"(X=A a S)"
```

```
1110 PRINT#1:PRINT#1,"pose le en";
1120 PRINT#1,"Y (Y=1 a 6)"
1130 PRINT#1,"range le"
1140 PRINT#1,"inventaire":PRINT#1,"ou est"
1150 PRINT#1,"et"
1160 PRINT#1,","
1170 xpince=2:GOSUB 2410
1180 GOTO 2720
1190 REM******************************************************************
1200 REM 2 eme partie : animation et modification des variables de l'univers
1210 REM******************************************************************
1220 REM==================================
1230 REM S/P deplacement a la position X,Y
1240 REM==================================
1250 IF x=0 THEN xdestination=2 ELSE xdestination=4+9*x
1260 ydestination=3*y+1
1270 FOR ypince=ypince TO 0 STEP -1
1280   GOSUB 2410
1290 NEXT ypince
1300 IF xdestination>xpince THEN sens=1 ELSE sens=-1
1310 FOR xpince=xpince TO xdestination STEP sens
1320   ypince=0:GOSUB 2410
1330 NEXT xpince
1340 xpince=xdestination
1350 FOR ypince=0 TO ydestination
1360   GOSUB 2410
1370 NEXT ypince
1380 ypince=ydestination
1390 RETURN
1400 REM================================
1410 REM saisie d'un objet dans la boite
1420 REM      entree : forme
1430 REM================================
1440 j=0
1450 IF forme.objet(j)<>forme OR existe(j)<>non THEN 1470
1460 cnpince$=lettre.objet$(j):GOTO 1550
1470 j=j+1
1480 IF j>nobjets THEN 1500
1490 GOTO 1450
1500 IF forme=cube THEN i=ncubes:ncubes=ncubes+1
1510 IF forme=ballon THEN i=nballons:nballons=nballons+1
1520 j=nobjets:nobjets=nobjets+1
1530 forme.objet(j)=forme
1540 lettre.objet$(j)=CHR$(65+i):cnpince$=CHR$(65+i)
1550 cfpince=forme:cipince=j
1560 existe(j)=oui
1570 GOSUB 2500
1580 REM--- animation ---
```

```
1590 x=0:y=1:GOSUB 1220
1600 pince$=LEFT$(pince$,102):ypince=5:GOSUB 2410
1610 pince$=LEFT$(pince$,79):ypince=6:GOSUB 2410
1620 pince$=LEFT$(pince$,60):ypince=7:GOSUB 2410
1630 pince$=p1$+f1$+CHR$(143):ypince=6:GOSUB 2410
1640 pince$=p1$+f1$+p2$+f2$+CHR$(212):ypince=5:GOSUB 2410
1650 pince$=p1$+f1$+p2$+f2$+p3$+f3$:ypince=4:GOSUB 2410
1660 pince$=pince$+p4$+esp7$:ypince=3:GOSUB 2410
1670 RETURN
1680 REM========================================
1690 REM rangement de l'objet tenu dans la boite
1700 REM========================================
1710 cfpince=vide
1720 existe(cipince)=non
1730 REM--- animation ---
1740 x=0:y=0:GOSUB 1220
1750 ypince=2:GOSUB 2410
1760 ypince=3:GOSUB 2410
1770 pince$=LEFT$(pince$,126):ypince=4:GOSUB 2410
1780 pince$=LEFT$(pince$,102):ypince=5:GOSUB 2410
1790 pince$=LEFT$(pince$,79):ypince=6:GOSUB 2410
1800 pince$=LEFT$(pince$,60):ypince=7:GOSUB 2410
1810 pince$=p1$+esp7$+CHR$(143):ypince=6:GOSUB 2410
1820 pince$=p1$+esp7$+p2$+esp7$+CHR$(212):ypince=5:GOSUB 2410
1830 o1$=esp7$:o2$=esp7$:o3$=esp9$:GOSUB 2380
1840 ypince=4:GOSUB 2410
1850 RETURN
1860 REM===========================
1870 REM  depose d'un objet en xd,yd
1880 REM===========================
1890 univers(xd,yd)=cfpince
1900 xobjet(cipince)=xd
1910 yobjet(cipince)=yd
1920 cfpince=vide
1930 REM--- animation ---
1940 x=xd:y=yd-1:GOSUB 1220
1950 pince$=LEFT$(pince$,126)
1960 FOR ypince=3*yd-1 TO 3*yd+1
1970   GOSUB 2410
1980 NEXT ypince
1990 o1$=esp7$:o2$=f1$:o3$=" "+f2$+" ":o4$=RIGHT$(f3$,8):GOSUB 2380
2000 ypince=ypince-2:GOSUB 2410
2010 o1$=esp7$:o2$=esp7$:o3$="":o4$="":GOSUB 2380
2020 ypince=ypince-1:GOSUB 2410
2030 LOCATE#2,xpince,ypince+10:PRINT#2," ";
2040 LOCATE#2,xpince+8,ypince+10:PRINT#2," ";
2050 o3$=esp9$:GOSUB 2380
2060 ypince=ypince-1:GOSUB 2410
2070 RETURN
```

```
2080 REM==========================
2090 REM saisie d'un objet en xd,yd
2100 REM==========================
2110 univers(xd,yd)=vide
2120 GOSUB 2620:cipince=i:cfpince=f
2130 forme=f:cnpince$=l$:GOSUB 2490
2140 REM--- animation ---
2150 x=xd:y=yd-1:GOSUB 1220
2160 pince$=LEFT$(pince$,102):ypince=ypince+1:GOSUB 2410
2170 o1$=esp7$:o2$=f1$:o3$=" "+f2$+" ":o4$=RIGHT$(f3$,8):GOSUB 2380
2180 ypince=ypince+1:GOSUB 2410
2190 o1$=f1$:o2$=f2$:o3$=f3$:o4$="":GOSUB 2380
2200 ypince=ypince+1:GOSUB 2410
2210 pince$=pince$+esp9$
2220 ypince=ypince-1:GOSUB 2410
2230 RETURN
2240 REM============================
2250 REM degagement de l'objet xe,ye
2260 REM============================
2270 FOR y=1 TO ye-1
2280   xd=xe:yd=y
2290   IF univers(xd,yd)=vide THEN 2320
2300   GOSUB 2110:REM saisie
2310   GOSUB 1710:REM rangement
2320 NEXT y
2330 RETURN
2340 REM==================
2350 REM petits S/P divers
2360 REM==================
2370 REM=== contenu de la pince ===
2380 pince$=p1$+o1$+p2$+o2$+p3$+o3$+p4$+o4$
2390 RETURN
2400 REM=== affichage du bras ===
2410 LOCATE#2,xpince,1
2420 LOCATE#2,xpince,1
2430 membre$(0)=coude$
2440 membre$(1)=bras$(ypince)+pince$
2450 FOR ii=0 TO 1
2460 FOR i=1 TO LEN(membre$(ii)) STEP 13:PRINT#2,MID$(membre$(ii),i,13);:NEXT
2470 NEXT ii
2480 RETURN
2490 REM=== forme de l'objet a manipuler ===
2500 IF forme=cube THEN f1$=c1$:f2$=c2$+cnpince$+c2$:f3$=c3$:RETURN
2510 IF forme=ballon THEN f1$=b1$:f2$=b21$+cnpince$+b22$:f3$=b3$:RETURN
2520 RETURN
2530 REM=== coordonnees de l'objet de forme f et de lettre l$ ===
2540 i=0
2550 IF forme.objet(i)=f AND lettre.objet$(i)=l$ AND existe(i)=oui THEN 2590
2560 IF i=nobjets THEN trouve=non:RETURN
```

```
2570 i=i+1
2580 GOTO 2550
2590 xp=xobjet(i):yp=yobjet(i):trouve=oui
2600 RETURN
2610 REM=== forme et lettre de l'objet de coordonnees xd,yd ===
2620 i=0
2630 IF xobjet(i)=xd AND yobjet(i)=yd AND existe(i)=oui THEN 2670
2640 IF i=nobjets THEN trouve=non:RETURN
2650 i=i+1
2660 GOTO 2630
2670 f=forme.objet(i):l$=lettre.objet$(i):trouve=oui
2680 RETURN
2690 REM***********************************************************
2700 REM  3 eme partie : gestion du dialogue et execution des ordres
2710 REM***********************************************************
2720 PRINT "Je suis a vos ordres"
2730 ph$="":LINE INPUT ph$
2740 ph$=LOWER$(ph$)
2750 REM--- decoupage de la phrase en ordres ---
2760 ordre$="":FOR k=1 TO LEN(ph$)
2770 car$=MID$(ph$,k,1)
2780   IF MID$(ph$,k,2)="et" THEN k=k+2:GOSUB 2850:GOTO 2810
2790   IF car$="," THEN GOSUB 2850:GOTO 2810
2800   ordre$=ordre$+MID$(ph$,k,1)
2810 NEXT k
2820 GOSUB 2850
2830 GOTO 2730
2840 REM--- decoupage de l'ordre en mots ---
2850 FOR j=0 TO nmots:mot$(j)="":NEXT j
2860 nmots=0
2870 FOR i=1 TO LEN(ph$)
2880   car$=MID$(ordre$,i,1)
2890   IF car$=" " THEN nmots=nmots+1:GOTO 2910
2900   mot$(nmots)=mot$(nmots)+MID$(ordre$,i,1)
2910 NEXT i
2920 PRINT "EXECUTION DE L'ORDRE : ";ordre$
2930 REM-------
2940 GOSUB 3000:REM analyse et execution d'un ordre
2950 ordre$=""
2960 RETURN
2970 REM================================
2980 REM analyse et execution d'un ordre
2990 REM================================
3000 j=0
3010 IF mot$(j)="prends" AND mot$(j+1)="le" THEN 3100
3020 IF mot$(j)="prends" AND mot$(j+1)="un" THEN 3190
3030 IF mot$(j)="pose" AND mot$(j+1)="le" AND mot$(j+2)="en" THEN 3280
3040 IF mot$(j)="range" AND mot$(j+1)="le" THEN 3420
3050 IF mot$(j)="inventaire" THEN 3460
```

```
3060 IF mot$(j)="ou" AND mot$(j+1)="est" THEN 3620
3070 IF j=nmots THEN 3990
3080 j=j+1:GOTO 3010
3090 REM--- prends le ---
3100 erreur=non:GOSUB 3830:IF erreur=oui THEN RETURN
3110 erreur=oui:REM en attendant la demonstration de l'inverse en 12180
3120 IF trouve=non THEN 4010
3130 IF i=cipince AND cfpince<>vide THEN 4040
3140 IF cfpince<>vide THEN GOSUB 1710
3150 xe=xp:ye=yp:GOSUB 2270
3160 xd=xp:yd=yp:GOSUB 2110
3170 erreur=non:RETURN
3180 REM--- prends un ---
3190 j=j+1
3200 IF mot$(j)="cube" THEN forme=cube:GOTO 3240
3210 IF mot$(j)="ballon" THEN forme=ballon:GOTO 3240
3220 IF j>nmots THEN 3990
3230 GOTO 3190
3240 IF cfpince<>vide THEN GOSUB 1710
3250 GOSUB 1440
3260 RETURN
3270 REM--- pose le sur ---
3280 j=j+3
3290 xd=VAL(mot$(j))
3300 IF xd>6 OR xd<1 THEN 4050
3310 IF cfpince=vide THEN 4060
3320 sommet=0
3330 FOR yd=1 TO 3
3340    IF univers(xd,yd)=vide THEN sommet=yd
3350 NEXT yd
3360 IF sommet=0 THEN 4070
3370 IF sommet=3 THEN 3390
3380 IF univers(xd,sommet+1)=ballon THEN 4080
3390 yd=sommet:GOSUB 1890
3400 RETURN
3410 REM--- range le ---
3420 IF mot$(j+2)<>"" OR cfpince=vide THEN 4060
3430 GOSUB 1710
3440 RETURN
3450 REM--- inventaire ---
3460 nb=0:nc=0
3470 IF cfpince=ballon THEN nb=1
3480 IF cfpince=cube THEN nc=1
3490 FOR xd=1 TO 6
3500    FOR yd=1 TO 3
3510       IF univers(xd,yd)=ballon THEN nb=nb+1
3520       IF univers(xd,yd)=cube THEN nc=nc+1
3530    NEXT yd
3540 NEXT xd
```

```
3550 IF nb+nc=0 THEN PRINT "Mon univers est vide":RETURN
3560 IF nc=0 THEN PRINT "Mon univers contient";nb;"ballons":RETURN
3570 IF nb=0 THEN PRINT "Mon univers contient";nc;"cubes":RETURN
3580 PRINT "Mon univers contient";nb+nc;"objets ";
3590 PRINT "dont";nb;"ballon(s) et";nc;"cube(s)"
3600 RETURN
3610 REM--- ou est ---
3620 erreur=non:GOSUB 3830:IF erreur=oui THEN RETURN
3630 IF trouve=non OR existe(i)=non THEN PRINT "Il est dans la boite":RETURN
3640 IF cipince=i AND cfpince=f THEN PRINT"Je le tiens":RETURN
3650 x=xobjet(i):y=yobjet(i)
3660 lieu$=""
3670 xd=x:yd=y+1:dir=0:GOSUB 3750
3680 xd=x+1:yd=y:dir=1:GOSUB 3750
3690 xd=x:yd=y-1:dir=2:GOSUB 3750
3700 xd=x-1:yd=y:dir=3:GOSUB 3750
3710 PRINT "Le ";forme$(fm);" ";lm$;" est en colonne";x
3720 IF lieu$="" THEN RETURN
3730 lieu$=LEFT$(lieu$,LEN(lieu$)-2)
3740 PRINT lieu$
3750 IF xd<1 OR xd>6 OR yd<1 OR yd>3 THEN RETURN
3760 IF univers(xd,yd)=vide THEN RETURN
3770 GOSUB 2620
3780 lieu$=lieu$+dir$(dir)+" "+forme$(f)+" "+l$+", "
3790 RETURN
3800 REM==========================
3810 REM analyse du nom d'un objet
3820 REM==========================
3830 j=j+1
3840 IF mot$(j)="cube" THEN f=cube:GOTO 3880
3850 IF mot$(j)="ballon" THEN f=ballon:GOTO 3880
3860 IF j>nmots THEN GOSUB 3990:erreur=oui:RETURN
3870 GOTO 3830
3880 j=j+1
3890 IF mot$(j)="" THEN GOSUB 3990:erreur=oui:RETURN
3900 IF LEN(mot$(j))>1 THEN GOSUB 4000:erreur=oui:RETURN
3910 mot$(j)=CHR$(ASC(mot$(j))-32)
3920 IF ASC(mot$(j))>90 OR ASC(mot$(j))<65 THEN GOSUB 4000:erreur=oui:RETURN
3930 l$=mot$(j):GOSUB 2540
3940 fm=f:lm$=l$
3950 RETURN
3960 REM==================
3970 REM messages d'erreur
3980 REM==================
3990 PRINT "Desole, je ne comprends pas cet ordre.":RETURN
4000 PRINT "Je ne donne jamais de tels noms a mes objets.":RETURN
4010 PRINT "Cette objet est dans la boite dans laquelle"
4020 PRINT "je parviens a peine a differencier les cubes des ballons,"
4030 PRINT "tant l'eclairage est mauvais.":RETURN
```

```
4040 PRINT "Je le tiens.":RETURN
4050 PRINT "Numero de colonne invalide.":RETURN
4060 PRINT "Impossible, je ne le tiens pas.":RETURN
4070 PRINT "Impossible, la colonne est pleine.":RETURN
4080 PRINT "Impossible, il ne tiendrait pas en equilibre":RETURN
```

PRÉSENTATION DU PROGRAMME SOURIS

COMMENT L'UTILISER ?

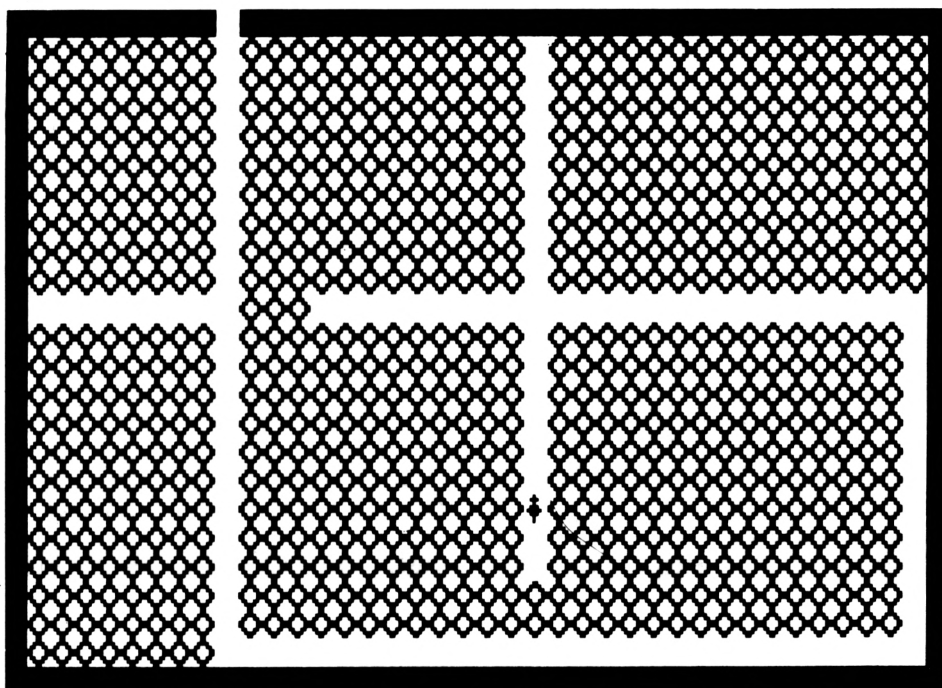
Le principal acteur de ce programme est une souris, perdue dans un labyrinthe, et qui recherche la sortie de manière intelligente, c'est-à-dire sans passer deux fois au même endroit, sauf quand cela accélère le processus d'exploration.

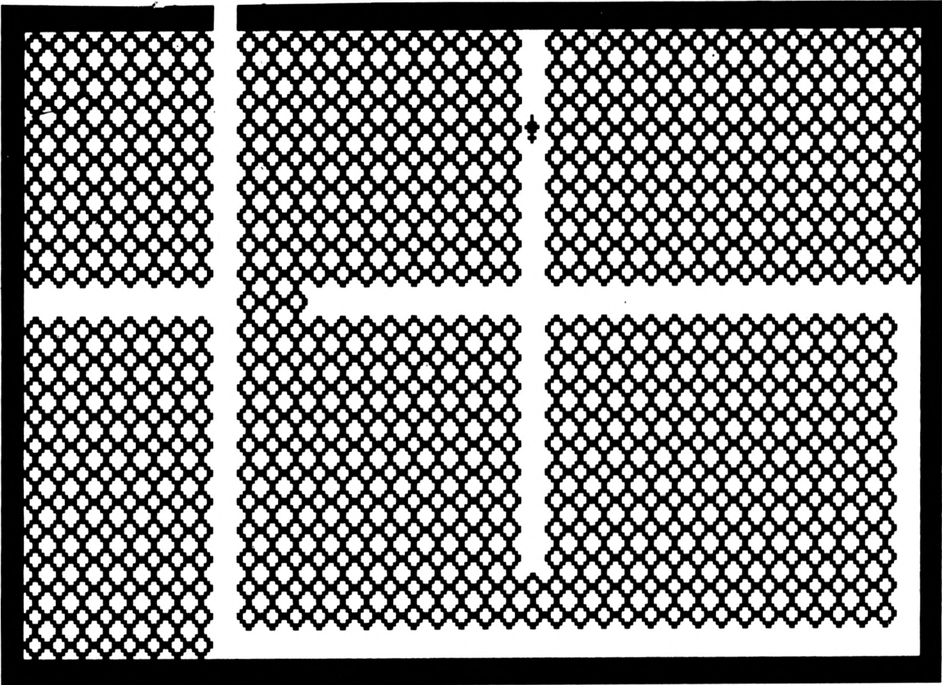
Votre rôle se décompose en trois phases :

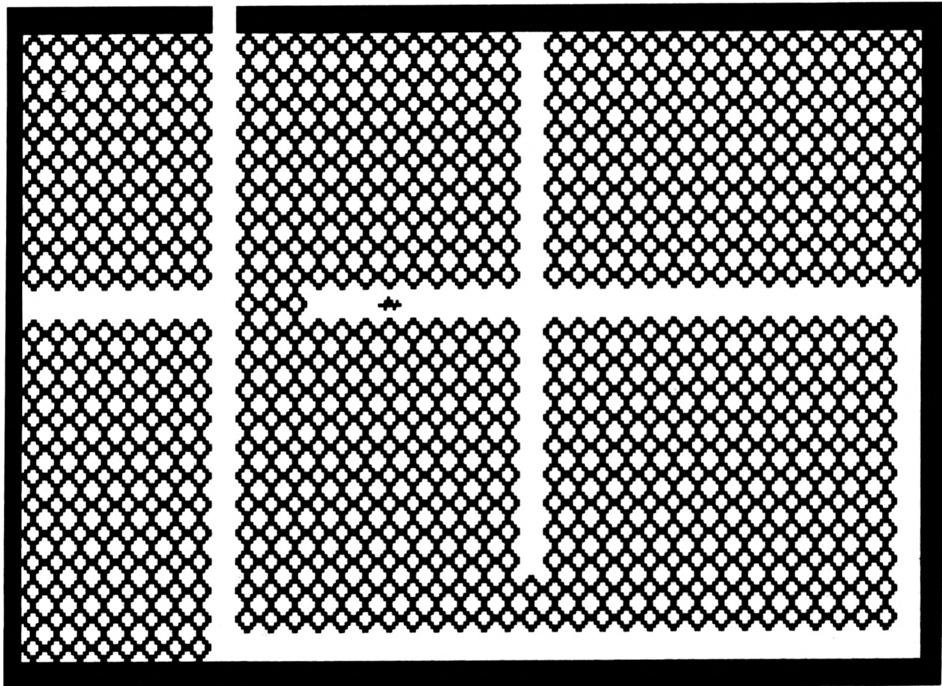
– choisir le lieu de la sortie sur le périmètre du labyrinthe en la déplaçant avec les flèches du curseur, puis en appuyant sur ENTER lorsque votre choix est effectué ;

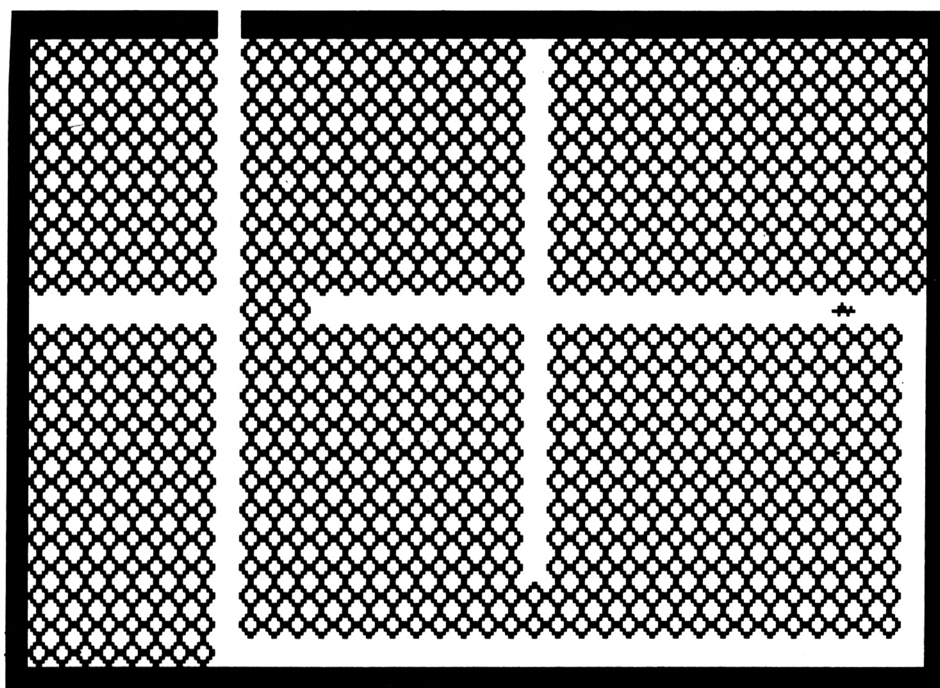
– dessiner un labyrinthe, toujours avec les flèches du curseur, mais cette fois en pressant simultanément la barre d'espace lorsque vous voulez "creuser" un chemin. Cette opération s'achève aussi par une pression sur ENTER ;

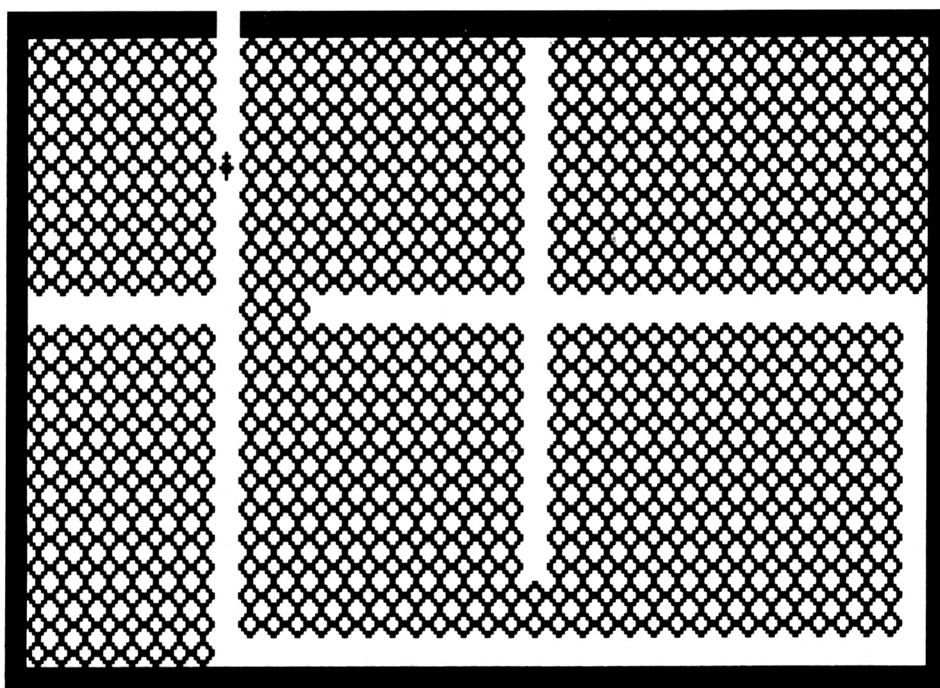
– enfin, placer la souris dans le labyrinthe, encore avec les flèches du curseur, puis la libérer, en pressant ENTER.











Dans ce labyrinthe assez simple, vous pouvez voir les différentes étapes de la recherche de la sortie, elle ne prendra qu'une dizaine de secondes.

COMMENT ÇA MARCHE ?

Ce logiciel comporte plusieurs modules destinés à initialiser les différentes variables et tableaux, ainsi que le terrain de jeu.

Ensuite deux modules permettent à l'utilisateur de créer le labyrinthe de son choix : le premier détermine la sortie, quant au second il permet de dessiner le labyrinthe, c'est une sorte de petit logiciel graphique.

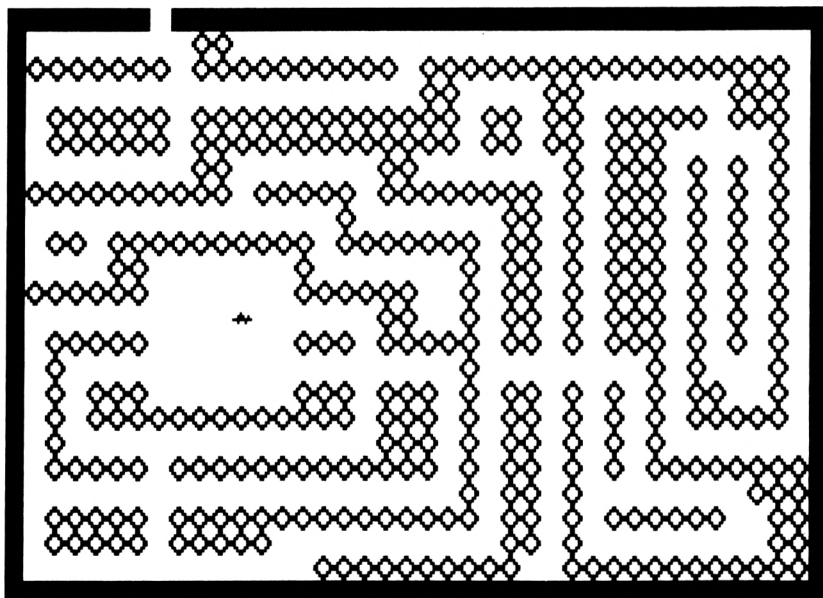
Lorsque toutes ces opérations sont effectuées, le cœur du logiciel prend le relais, il s'agit du module recherchant la sortie du labyrinthe.

Ce module fonctionne selon un algorithme récursif : il s'appelle lui-même, et donc s'enfonce à chaque appel d'un niveau de profondeur dans l'exploration, du moins, lorsqu'il peut aller dans une case qu'il n'a pas encore explorée, mémorisant dans une pile la trace de son passage. A l'inverse, il remonte dans l'exploration dès qu'il rencontre une case déjà explorée, dépilant la pile et revenant à la case indiquée par le terme dépilé. A force de dépiler, le programme revient à une intersection à partir de laquelle il repart dans une direction non explorée, le module recommence alors à s'appeler lui-même.

La souris se comporte finalement comme un petit poucet qui laisse des miettes de pain derrière lui. A chaque fois qu'elle rencontre devant elle des miettes, elle fait marche arrière, jusqu'à ce qu'elle trouve une bifurcation non explorée. Notre souris recherche la sortie tel que le ferait un être humain qui n'aurait pas le plan du labyrinthe.

La souris ne passe donc jamais deux fois au même endroit. A cette règle, une exception : si elle en a la possibilité, elle passera jusqu'à deux fois par une intersection, ce qui lui fera gagner beaucoup de temps dans la majeure partie des cas. Autre perfectionnement, lorsque la souris a le choix entre plusieurs chemins elle "préfère" celui qui lui fait garder son cap.

Le programme explore en fait un arbre dont les nœuds sont les bifurcations du labyrinthe, et dont les branches sont les tronçons qui rejoignent ces intersections.



Ce labyrinthe est la preuve qu'aucun piège et aucun cas de figure n'empêchera la souris de trouver la sortie. A vous de créer les labyrinthes les plus compliqués !

PRINCIPALES VARIABLES

case(i)	état de la case i (explorée, inexplorée, vide, pleine, bord)
chemin(i)	mémorisation des cases explorées à partir de la dernière intersection

```
10 REM*****************
20 REM SOURIS (T. L.-A)
30 REM*****************
40 MODE 2:INK 0,13:INK 1,0
50 PRINT"     MODE D'EMPLOI"
60 PRINT
70 PRINT" L'utilisation de ce logiciel s'effectue en 3 phases :"
80 PRINT
90 PRINT"1:  Choisissez la sortie sur le perimetre du labyrinthe"
100 PRINT"   avec les fleches de curseur puis appuyez sur la touche ENTER."
110 PRINT
120 PRINT"2:  Dessinez un labyrinthe avec les fleches du curseur"
130 PRINT"  et la touche ESPACE pressee simultanement pour creuser un chemin."
140 PRINT"    Appuyez sur ENTER pour terminer cette phase."
150 PRINT
160 PRINT"3:  Placez la souris ou bon vous semble mais dans une case vide""
170 PRINT"  avec les fleches du curseur."
180 PRINT"    Pressez la touche ENTER pour finir.""
190 PRINT
200 INPUT"Pour continuer, appuyez sur ENTER ",rep$
210 MODE 1:INK 0,1:INK 1,24
220 WINDOW 1,40,1,24
230 WINDOW#1,1,40,25,25
240 REM=========================================
250 REM initialisation des tableaux et variables
260 REM=========================================
270 DEFINT a-z
280 DIM case(960),dep(4),sym(3),chemin(960),mdir(960)
290 DIM souris$(3)
300 non.exploree=0
310 exploree=1
320 mur=2
330 sortie=3
340 bord=4
350 coin=5
360 intersection=6
370 nord=0:est=1:sud=2:ouest=3:arret=4
380 sym(nord)=sud:sym(sud)=nord:sym(est)=ouest:sym(ouest)=est
```

```
390 dep(nord)=-40
400 dep(est)=1
410 dep(sud)=40
420 dep(ouest)=-1
430 dep(arret)=0
440 baisse=1:leve=0
450 REM---graphisme de la souris---
460 SYMBOL 245,16,56,16,56,124,56,16,16
470 SYMBOL 246,0,0,16,58,255,36,0,0
480 SYMBOL 247,16,16,56,124,56,16,56,16
490 SYMBOL 248,0,0,8,92,255,36,0,0
500 souris$(nord)=CHR$(245)
510 souris$(est)=CHR$(246)
520 souris$(sud)=CHR$(247)
530 souris$(ouest)=CHR$(248)
540 REM===========================
550 REM initialisation du terrain
560 REM===========================
570 REM---murs partout---
580 FOR pcase=1 TO 960
590   case(pcase)=mur
600 NEXT pcase
610 FOR i=1 TO 24:PRINT STRING$(40,202);:NEXT i
620 REM---cadre---
630 FOR x=1 TO 40
640   LOCATE x,1:PRINT CHR$(143);:case(x)=bord
650   LOCATE x,24:PRINT CHR$(143);:case(x+920)=bord
660 NEXT x
670 FOR y=2 TO 23
680   LOCATE 1,y:PRINT CHR$(143);:case(40*(y-1)+1)=bord
690   LOCATE 40,y:PRINT CHR$(143);:case(40*y)=bord
700 NEXT y
710 case(1)=coin:case(40)=coin:case(921)=coin:case(960)=coin
720 REM====================
730 REM choix de la sortie
740 REM====================
750 PRINT#1,"choisissez la sortie (fleches et enter)"
760 pcase=1:xm=1:ym=1
770 LOCATE 1,1:PRINT " ";
780 WHILE INKEY$<>CHR$(13)
790   FOR temps=1 TO 200:NEXT temps
800   GOSUB 1820:REM test du clavier
810   IF pcase<1 THEN pcase=960+pcase
820   IF pcase>960 THEN pcase=pcase-960
830   IF case(pcase)<>bord AND case(pcase)<>coin THEN pcase=pcase-dep(dir)
840   GOSUB 1930:REM conversion de pcase en x et y
850   IF x=xm AND y=ym THEN 890
860   LOCATE x,y:PRINT CHR$(32);
870   LOCATE xm,ym:PRINT CHR$(143);
```

```
880   xm=x:ym=y
890 WEND
900 IF case(pcase)=coin THEN 780
910 case(pcase)=sortie
920 REM====================
930 REM dessin du labyrinthe
940 REM====================
950 PRINT#1,"Dessinez un labyrinthe avec"
960 FOR temps=1 TO 4000:NEXT temps
970 PRINT#1,"Les FLECHES et ESPACE, puis ENTER"
980 pcase=42:pmcase=42:xm=2:ym=2
990 LOCATE 2,2:PRINT CHR$(144);
1000 WHILE INKEY$<>CHR$(13)
1010   GOSUB 1820:REM test du clavier
1020   FOR temps=1 TO 200:NEXT temps
1030   IF INKEY(47)=0 THEN crayon=baisse ELSE crayon=leve
1040   IF case(pcase)=bord OR case(pcase)=sortie THEN pcase=pcase-dep(dir)
1050   IF crayon=baisse THEN case(pcase)=vide
1060   IF pcase=pmcase THEN 1120
1070   GOSUB 1930:REM conversion de pcase en x et y
1080   LOCATE x,y:PRINT CHR$(144);
1090   LOCATE xm,ym
1100   IF case(pmcase)=mur THEN PRINT CHR$(202); ELSE PRINT CHR$(32);
1110   pmcase=pcase:xm=x:ym=y
1120 WEND
1130 LOCATE x,y
1140 IF case(pcase)=mur THEN PRINT CHR$(202); ELSE PRINT " ";
1150 REM=====================================
1160 REM CHOIX DU POINT DE DEPART DE LA SOURIS
1170 REM=====================================
1180 PRINT#1,"Placez la souris dans le labyrinthe"
1190 FOR temps=1 TO 4000:NEXT temps
1200 PRINT#1,"Avec les fleches et la touche ENTER"
1210 psouris=42:pmsouris=42:xm=2:ym=2
1220 LOCATE 2,2:PRINT CHR$(144);
1230 WHILE INKEY$<>CHR$(13)
1240   pcase=psouris:GOSUB 1820:psouris=pcase:REM test clavier
1250   FOR temps=1 TO 200:NEXT temps
1260   IF case(psouris)=bord OR case(psouris)=sortie THEN psouris=psouris-dep(di
r)
1270   IF psouris=pmsouris THEN 1320
1280   p$=souris$(dir):GOSUB 1990:REM affichage de la souris
1290   LOCATE xm,ym
1300   IF case(pmsouris)=mur THEN PRINT CHR$(202); ELSE PRINT " ";
1310   pmsouris=psouris:xm=x:ym=y
1320 WEND
1330 IF case(pmsouris)=mur THEN 1230
1340 dir=0:CLS#1
1350 REM************************
```

```
1360 REM RECHERCHE DE LA SORTIE
1370 REM************************
1380 ndir=0
1390 FOR essai.dir=nord TO ouest
1400   IF essai.dir=mdir THEN 1420
1410   GOSUB 1610
1420 NEXT essai.dir
1430 essai.dir=mdir:GOSUB 1610
1440 IF ndir=0 THEN 1550
1450 IF ndir>2 THEN case(psouris)=intersection
1460 IF ndir=1 THEN case(psouris)=exploree
1470 p$=" ":GOSUB 1990
1480 psouris=psouris+dep(dir)
1490 p$=souris$(dir):GOSUB 1990
1500 GOSUB 1680
1510 REM memorisation du chemin
1520 chemin(pt)=psouris:mdir(pt)=sym(dir):pt=pt+1
1530 IF case(psouris)<>intersection THEN case(psouris)=exploree
1540 GOTO 1380
1550 p$=" ":GOSUB 1990
1560 IF pt=0 THEN 1730:REM fin sans solution
1570 pt=pt-1:psouris=chemin(pt):REM retour sur ses pas
1580 p$=souris$(mdir(pt)):GOSUB 1990
1590 GOTO 1380
1600 REM--- examen des cases environnantes ---
1610   case=case(psouris+dep(essai.dir))
1620   IF case=non.exploree OR case=sortie THEN ndir=ndir+1:dir=essai.dir
1630   IF case=intersection AND case(psouris)<>intersection THEN ndir=ndir+1:dir
=essai.dir
1640 RETURN
1650 REM--------------------------
1660 REM TEST DE FIN DE RECHERCHE
1670 REM--------------------------
1680 IF case(psouris)=sortie THEN 1740:REM fin avec solution
1690 RETURN
1700 REM==================
1710 REM  fin de recherche
1720 REM==================
1730 PRINT#1,"Pas de sortie, on recommence (o ou n)?":GOTO 1750
1740 PRINT#1,"On recommence (o ou n)?"
1750 rep$=INKEY$
1760 IF rep$="o" OR rep$="O" THEN RUN
1770 IF rep$="n" OR rep$="N" THEN MODE 1:END
1780 GOTO 1750
1790 REM========================================
1800 REM TEST AU CLAVIER DE LA DIRECTION CHOISIE
1810 REM========================================
1820 dir=arret
1830 IF INKEY(0)=0 THEN dir=nord
```

```
1840 IF INKEY(1)=0 THEN dir=est
1850 IF INKEY(2)=0 THEN dir=sud
1860 IF INKEY(8)=0 THEN dir=ouest
1870 pcase=pcase+dep(dir)
1880 RETURN
1890 REM========================================
1900 REM TRANSFORMATION DU NUMERO DE CASE  pcase
1910 REM         EN COORDONNEES x ET y
1920 REM========================================
1930 y=1+INT((pcase-1)/40):x=pcase-40*(y-1)
1940 RETURN
1950 REM=======================
1960 REM AFFICHAGE OU EFFACEMENT
1970 REM      DE LA SOURIS
1980 REM=======================
1990 PCASE=PSOURIS
2000 GOSUB 1930:REM conversion de pcase en x et y
2010 LOCATE x,y:PRINT p$
2020 adir=dir
2030 RETURN
```

SOPHIE : BOUCHE ANIMÉE

COMMENT L'UTILISER ?

Ce logiciel s'utilise par l'intermédiaire d'un menu offrant trois options :

– entrée d'une phrase, formée en l'occurrence d'une séquence de phonèmes (dont la liste figure en bas de l'écran), séparés par des points (.) ;

– modification de la vitesse ;

– exécution de la séquence. Cette option provoque l'affichage de Sophie, puis l'animation de sa bouche, en accord avec la phrase frappée.

COMMENT ÇA MARCHE ?

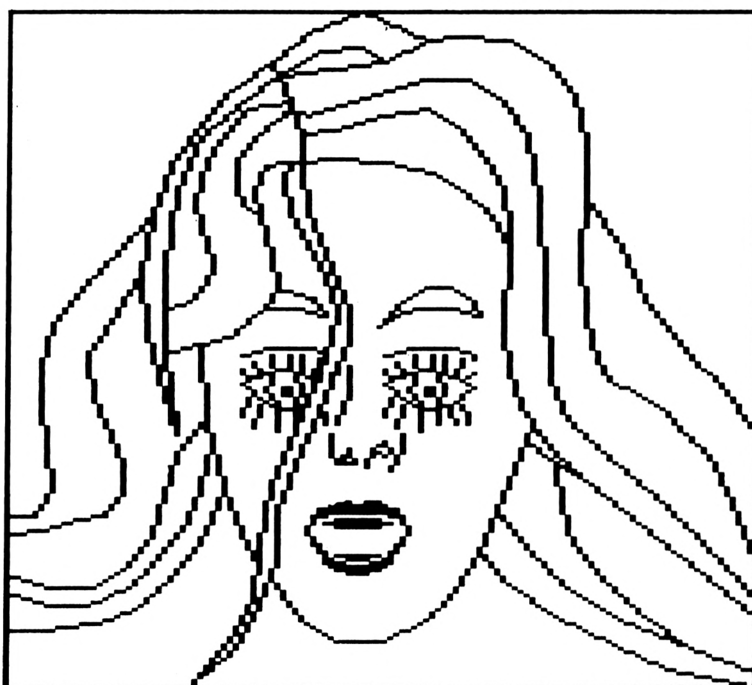
Le fonctionnement de ce programme est des plus simples. Passons sur l'affichage du portrait, qui est élémentaire, pour nous attarder sur l'animation de la bouche, en fonction du texte frappé.

A chaque phonème est associé le dessin d'une bouche correspondant à la prononciation de ce phonème.

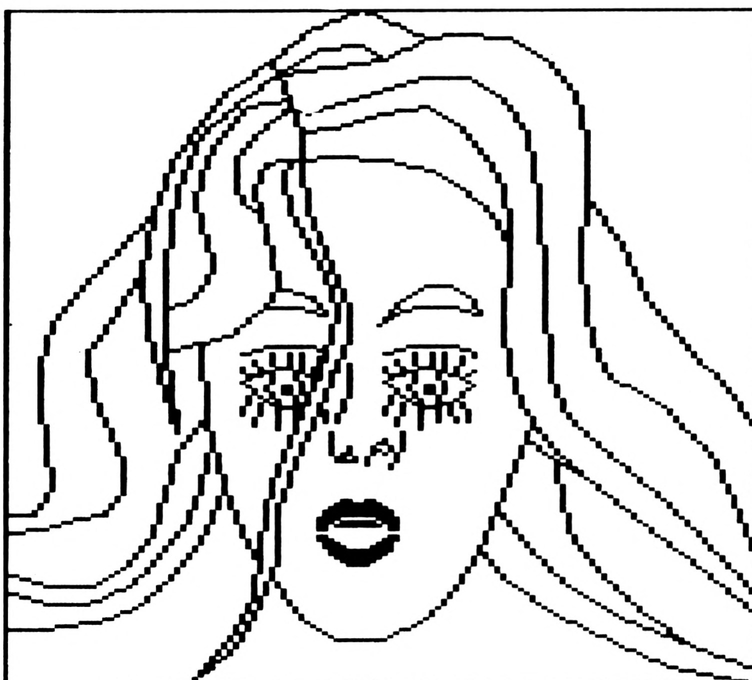
Le texte frappé par l'utilisateur est converti, phonème par phonème, en code numérique, et placé dans un tableau.

Lorsque le texte est prononcé, chaque code est pris un par un et le dessin correspondant est affiché. Les phonèmes correspondant à des consonnes sont affichés cinq fois moins longtemps.

Sophie prononce la séquence : a.e.i.o.u



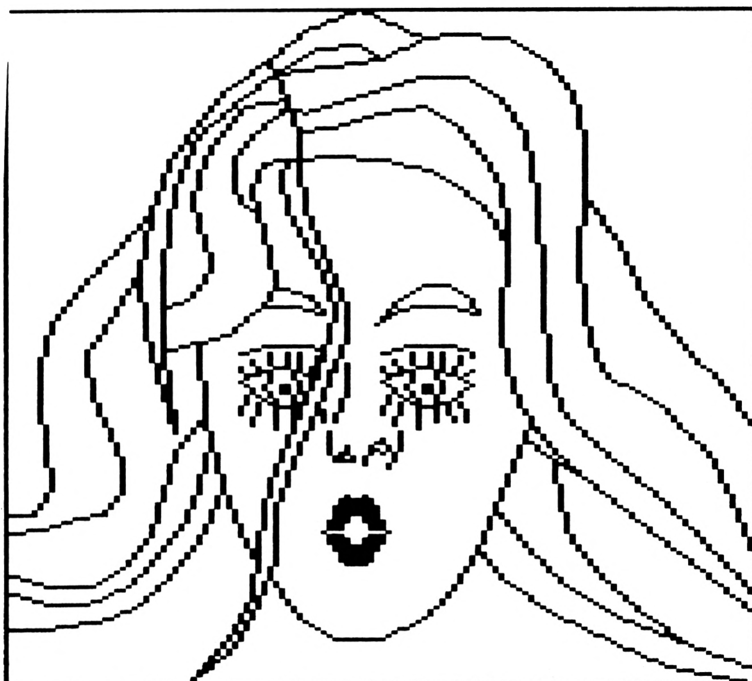
Sophie 1 : a



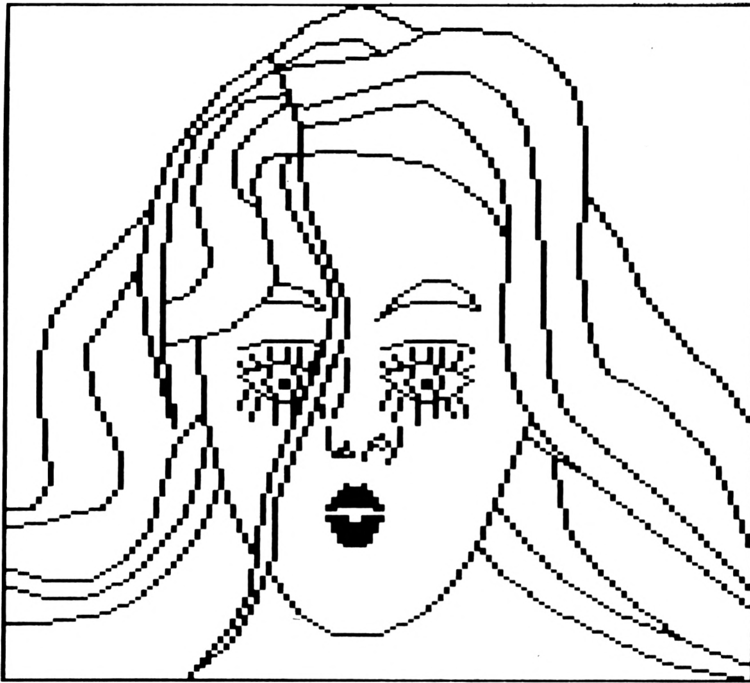
Sophie 2 : e



Sophie 3 : i



Sophie 4 : o



Sophie 5 : u

PRINCIPALES VARIABLES

phon$(i,j)	j ième chaîne graphique du phonème de forme i de bouche
pho$(i)	phonétique du phonème i
sequence(i)	i ème phonème de la séquence à prononcer
iphon(i)	forme de bouche du phonème i

```
10 REM*******************
20 REM   SOPHIE (T. L.-A)
30 REM*******************
40 REM===============
50 REM initialisation
60 REM===============
70 MODE 2:PRINT "Un instant ..."
80 DEFINT a-z:DEFREAL x,y,p,v
90 consomne=1:p=4
100 DIM phon$(10,3),x(230),y(230),pho$(27),iphon(27),sequence(300)
110 GOSUB 870:REM definition des chaines de caracteres graphiques et points
120 REM============== MENU ===============
130 MODE 2
140 WINDOW 1,80,1,22:WINDOW#2,1,80,24,25
```

```
150 FOR i=0 TO 27:PRINT#2,pho$(i);".";:NEXT i
160 LOCATE#3,1,23:PRINT#3,STRING$(80,233);
170 PRINT "Entree d'une nouvelle sequence --------> 1"
180 PRINT "Modification de la vitesse ------------> 2"
190 PRINT "Execution de la sequence en memoire ---> 3"
200 INPUT rep$:rep$=LOWER$(rep$)
210 IF rep$="1" THEN 260
220 IF rep$="2" THEN 480
230 IF rep$="3" THEN 610
240 PRINT "Repondez par 1,2 ou 3":GOTO 200
250 REM==== ordre 1 du menu ====
260 PRINT "Entrez une sequence de phonemes (voir liste ci-dessous)"
270 PRINT "en les separant par des points ('.')"
280 INPUT ph$:IF RIGHT$(ph$,1)<>"." THEN ph$=ph$+"."
290 REM==codage des phonemes
300 k=0
310 FOR i=1 TO LEN(ph$)
320   i$=MID$(ph$,i,1)
330   IF i$<>"." THEN 440
340   REM--- codage d'un phoneme
350   j=0
360   IF pho$(j)=phon$ THEN sequence(k)=iphon(j):k=k+1:GOTO 420
370   IF j<>27 THEN j=j+1:GOTO 360
380   REM--- correction
390   PRINT "Le phoneme ";phon$;" n'existe pas"
400   INPUT "Par quel phoneme voulez-vous le remplacer";phon$:GOTO 350
410   REM---
420   phon$="":GOTO 450
430   REM---
440   phon$=phon$+i$
450 NEXT i
460 np=k
470 CLS:GOTO 170
480 REM=== ordre 2 du menu ===
490 CLS
500 IF p=0 THEN v=100:GOTO 520
510 v=100*SQR(1/p)
520 PRINT "La vitesse est de";v;"%"
530 INPUT "quelle est la nouvelle valeur";v
540 IF v<0 OR v>100 THEN PRINT "entre 0 et 100 %, SVP":GOTO 530
550 IF v=0 THEN PRINT "Valeur qui conduirait a une lenteur...infinie!":GOTO 530
560 IF v=100 THEN p=0:CLS:GOTO 170
570 IF v=0 THEN p=
580 p=10000/(v*v)
590 CLS:GOTO 170
600 REM=== ordre 3 du menu ===
610 IF np=0 THEN CLS:PRINT "Il n'y a pas de sequence en memoire":GOTO 170
620 GOSUB 2300:REM affichage du visage
630 WINDOW 1,20,1,1:WINDOW#2,13,20,1,25
```

```
640 LOCATE#1,7,19:PRINT#1,phon$(0,1);
650 LOCATE#1,7,20:PRINT#1,phon$(0,2);
660 LOCATE#1,7,21:PRINT#1,phon$(0,3);
670 FOR j=1 TO 1000:NEXT j
680 FOR i=0 TO np-1
690   IF sequence(i)=10 THEN FOR j=1 TO 100*p:NEXT j
700   LOCATE#1,7,19:PRINT#1,phon$(sequence(i),1);
710   LOCATE#1,7,20:PRINT#1,phon$(sequence(i),2);
720   LOCATE#1,7,21:PRINT#1,phon$(sequence(i),3);
730   IF l(sequence(i))=voyelle THEN FOR j=1 TO 100*p:NEXT j:GOTO 750
740   FOR j=1 TO 15*p:NEXT j
750 NEXT i
760 LOCATE#1,7,19:PRINT#1,phon$(0,1);
770 LOCATE#1,7,20:PRINT#1,phon$(0,2);
780 LOCATE#1,7,21:PRINT#1,phon$(0,3);
790 CLS:INPUT "On recommence";rep$:CLS
800 rep$=LEFT$(LOWER$(rep$),1)
810 IF rep$="o" THEN 680
820 IF rep$="n" THEN 130
830 GOTO 790
840 REM====================================
850 REM definition des caracteres graphiques
860 REM====================================
870 SYMBOL AFTER 123
880 ret$=CHR$(10)+STRING$(3,8)
890 SYMBOL 123,0,0,0,0,0,0,0,102
900 SYMBOL 124,0,1,7,15,0,15,7,3
910 SYMBOL 125,255,255,255,255,0,255,255,255
920 SYMBOL 126,0,128,224,240,0,240,224,192
930 SYMBOL 127,255,126,24,0,0,0,0,0
940 SYMBOL 128,1,0,0,0,0,0,0,0
950 SYMBOL 129,128,0,0,0,0,0,0,0
960 REM----------------------
970 SYMBOL 130,0,0,0,0,64,64,129,135
980 SYMBOL 131,0,0,0,0,0,0,195,231
990 SYMBOL 132,0,0,0,0,2,2,129,255
1000 SYMBOL 133,191,167,144,155,73,76,78,71
1010 SYMBOL 134,255,255,0,255,255,0,255,60
1020 SYMBOL 135,253,229,9,217,18,50,114,226
1030 SYMBOL 136,67,1,0,0,0,0,0,0
1040 SYMBOL 137,195,255,255,60,0,0,0,0
1050 SYMBOL 138,194,128,0,0,0,0,0,0
1060 REM----------------------
1070 SYMBOL 139,0,0,36,126,126,255,255,255
1080 SYMBOL 140,0,1,1,1,2,1,1,1
1090 SYMBOL 141,231,231,195,195,0,195,231,231
1100 SYMBOL 142,0,128,128,128,64,128,128,128
1110 SYMBOL 143,255,255,126,126,126,60,0,0
1120 SYMBOL 144,0,0,0,0,1,7,7,14
```

```
1130 REM----------------------
1140 SYMBOL 145,0,0,0,231,255,255,255,0
1150 SYMBOL 146,0,0,0,0,128,224,224,112
1160 SYMBOL 147,8,11,16,16,16,8,8,12
1170 SYMBOL 148,255,255,255,0,0,0,0,0
1180 SYMBOL 149,16,208,8,8,8,16,16,48
1190 SYMBOL 150,4,6,2,3,3,1,0,0
1200 SYMBOL 151,0,0,255,60,195,255,255,60
1210 SYMBOL 152,32,96,64,192,192,128,0,0
1220 REM---------------------
1230 SYMBOL 153,0,0,0,0,0,0,1,1
1240 SYMBOL 154,36,60,126,126,255,255,255,129
1250 SYMBOL 155,0,0,0,0,0,0,128,128
1260 SYMBOL 156,0,1,1,0,0,0,0,0
1270 SYMBOL 157,0,231,231,255,255,255,126,126
1280 SYMBOL 158,0,128,128,0,0,0,0,0
1290 SYMBOL 159,126,60,0,0,0,0,0,0
1300 REM--------------------
1310 SYMBOL 160,0,0,0,0,0,1,3,3
1320 SYMBOL 161,0,0,0,102,255,255,255,255
1330 SYMBOL 162,0,0,0,0,0,128,192,192
1340 SYMBOL 163,7,7,6,6,0,6,7,3
1350 SYMBOL 164,129,126,255,0,0,0,0,129
1360 SYMBOL 165,224,224,96,96,0,96,224,192
1370 SYMBOL 166,3,1,1,0,0,0,0,0
1380 SYMBOL 167,195,255,255,255,126,60,0,0
1390 SYMBOL 168,192,192,128,0,0,0,0,0
1400 REM--------------------
1410 SYMBOL 170,0,0,0,0,0,0,0,1
1420 SYMBOL 171,0,0,0,36,60,126,255,255
1430 SYMBOL 172,0,0,0,0,0,0,0,128
1440 SYMBOL 173,3,3,3,7,0,7,3,3
1450 SYMBOL 174,195,189,126,0,0,0,195,255
1460 SYMBOL 175,192,192,192,224,0,224,192,192
1470 SYMBOL 176,1,0,0,0,0,0,0,0
1480 SYMBOL 177,255,255,126,60,0,0,0,0
1490 SYMBOL 178,128,0,0,0,0,0,0,0
1500 REM--------------------
1510 SYMBOL 179,0,0,0,7,8,7,33,64
1520 SYMBOL 180,0,36,255,255,0,255,255,126
1530 SYMBOL 181,0,0,0,224,16,224,132,2
1540 SYMBOL 182,64,128,128,128,0,0,0,0
1550 SYMBOL 183,2,1,1,1,0,0,0,0
1560 REM--------------------
1570 SYMBOL 184,0,0,0,0,0,0,0,102
1580 SYMBOL 185,0,1,3,7,15,0,15,7
1590 SYMBOL 186,255,255,255,195,129,66,231,255
1600 SYMBOL 187,0,128,192,224,240,0,240,224
1610 SYMBOL 188,7,3,1,0,0,0,0,0
```

```
1620 SYMBOL 189,255,255,255,60,0,0,0,0
1630 SYMBOL 190,224,192,128,0,0,0,0,0
1640 REM--------------------
1650 SYMBOL 191,0,36,126,126,255,231,219,189
1660 SYMBOL 192,1,1,1,1,1,1,1,1
1670 SYMBOL 193,129,129,0,0,0,0,129,129
1680 SYMBOL 194,128,128,128,128,128,128,128,128
1690 SYMBOL 195,189,219,219,231,126,126,60,24
1700 REM=== forme des yeux ===
1710 SYMBOL 240,15,48,196,4,36,36,147,156
1720 SYMBOL 241,248,6,161,165,164,164,168,105
1730 SYMBOL 242,116,68,132,73,41,37,84,74
1740 SYMBOL 243,90,68,34,161,162,164,68,72
1750 SYMBOL 244,86,147,146,146,34,34,34,2
1760 SYMBOL 245,116,212,82,82,74,74,0,0
1770 DATA 240,241,10,8,8,242,243,10,8,8,244,245
1780 FOR i=1 TO 12:READ a:oeil$=oeil$+CHR$(a):NEXT i
1790 REM===================================
1800 REM definition de la forme de la bouche
1810 REM pour chaque phoneme
1820 REM===================================
1830 REM--- repos ---
1840 DATA 32,123,32,124,125,126,32,127,32
1850 REM--- I ---
1860 DATA 130,131,132,133,134,135,136,137,138
1870 REM--- O ---
1880 DATA 32,139,32,140,141,142,32,143,32
1890 REM--- A ---
1900 DATA 144,145,146,147,148,149,150,151,152
1910 REM--- U ---
1920 DATA 153,154,155,156,157,158,32,159,32
1930 REM--- E ---
1940 DATA 160,161,162,163,164,165,166,167,168
1950 REM--- C,S,D,J,G,K,L,N,R,T ---
1960 DATA 170,171,172,173,174,175,176,177,178
1970 REM--- B,M,P ---
1980 DATA 32,32,32,179,180,181,182,32,183
1990 REM--- F,V ---
2000 DATA 32,184,32,185,186,187,188,189,190
2010 REM--- en ---
2020 DATA 32,191,32,192,193,194,32,195,32
2030 REM=== remplissage des variables associees a chaque phoneme ===
2040 FOR i=0 TO 9
2050   FOR j=1 TO 3
2060     FOR k=1 TO 3
2070       READ a
2080       phon$(i,j)=phon$(i,j)+CHR$(a)
2090     NEXT k
2100   NEXT j
```

```
2110 NEXT i
2120 DATA repos,0, i,1, e/,1, ou,2, o,2, on,2, a,3, in,3, eç,3, u,4, e,5, z,6
2130 DATA s,6, d,6, j,6, g,6, k,6, l,6, n,6, r,6, t,6, b,7,m,7, p,7, f,8, v,8
2140 DATA en,9, fige,10
2150 FOR i=0 TO 27
2160   READ a$,b
2170   pho$(i)=a$
2180   iphon(i)=b
2190 NEXT i
2200 FOR i=6 TO 8:l(i)=consomne:NEXT i
2210 REM--- remplissage des points ---
2220 FOR i=1 TO 227
2230   READ x,y
2240   x(i)=x:y(i)=y
2250 NEXT i
2260 RETURN
2270 REM=================
2280 REM dessin du visage
2290 REM==================
2300 echx=27:echy=27:x0=3:y0=0:d$="saut"
2310 MODE 0
2320 LOCATE 6,14:PRINT oeil$
2330 LOCATE 9,14:PRINT oeil$
2340 RESTORE 3020
2350 READ a$
2360 IF a$="fin" THEN 2440
2370 IF a$="f" THEN d$="saut":GOTO 2350
2380 pt=VAL(a$)
2390 x=echx*x(pt)+x0
2400 y=echy*y(pt)+y0
2410 IF d$="saut" THEN MOVE x,y:d$="trace":GOTO 2350
2420 IF d$="trace" THEN DRAW x,y,1:GOTO 2350
2430 REM === retouches ===
2440 PLOT 220,187,0:PLOT 220,185:PLOT 216,175:PLOT 216,171:PLOT 212,165
2450 PLOT 212,163:PLOT 208,161:PLOT 204,157:PLOT 204,155:PLOT 204,153
2460 RETURN
2470 REM=== coordonnees des points ===
2480 DATA 8.2,1, 9.4,1, 10.5,1.4, 11.2,2, 11.8,2.8
2490 DATA 12.6,2.2, 13.7,1.5, 15.2,0.9, 16.5,0.4, 18.7,0
2500 DATA 18.7,0.3, 17.2,0.8, 15.3,1.4, 14,2.1, 13,2.9
2510 DATA 12.2,3.7, 16,1.5, 15,2.2, 14.2,3, 13.9,3.8
2520 DATA 13.7,4.7, 13,5.1, 18.7,1.5, 14.5,4.3, 13.6,5
2530 DATA 13.1,5.5, 12.8,5.9, 12.5,6.6, 12.4,7.4, 12.5,9.1
2540 DATA 12.6,9.5, 12.4,10.3, 12,11, 11.4,11.6, 10.5,12
2550 DATA 9.4,11.9, 8.3,11.6, 7.3,11.5, 12.1,9.8, 11.4,10.3
2560 DATA 10.6,10.7, 9.4,10.9, 8.3,11, 7.4,10.9, 7.1,10.9
2570 DATA 6.5,10.8, 6.2,10.3, 6.3,9.8, 6.6,9, 6.7,8.3
2580 DATA 6.4,7.9, 6.2,7.7, 5.5,7.3, 4.9,7.1, 4,6.9
2590 DATA 4.1,6.3, 4.3,5.3, 3.9,6.1, 3.6,7, 3.5,7.4
```

```
2600 DATA 3.4,8.6, 3.6,9.8, 4.1,10.8, 4.8,11.7, 5.6,12.4
2610 DATA 6.6,13, 7.5,13.6, 8.8,14, 10.5,13.4, 11.4,13.5
2620 DATA 12.5,13.4, 13.4,13, 14.1,12.3, 14.4,11.3, 14.4,10.2
2630 DATA 15.3,9.1, 15.9,8.1, 16.4,7.2, 17.1,6.8, 17.8,6.3
2640 DATA 18.7,5.3, 14.3,8.9, 14.4,7.9, 14.6,7.1, 15.1,6.4
2650 DATA 15.8,6.1, 16.4,5.8, 17,5.3, 17.6,4.8, 18.1,3.9
2660 DATA 18.7,2.7, 18.7,1.9, 17.1,3.3, 16.4,3.9, 14.9,4.9
2670 DATA 14,5.6, 13.6,6.4, 13.3,10.1, 13,11.4, 12.4,12.2
2680 DATA 11.5,12.6, 10.4,12.6, 8.3,12.1, 7.5,11.9
2690 DATA 7,12, 6.1,11.4, 5.7,10.9, 5.7,10.4
2700 DATA 5.8,10, 9.4,13, 8.8,13.3, 7.8,13.2
2710 DATA 6.8,12.7, 7,12.2, 6,12, 5.2,11.2
2720 DATA 4.8,10.5, 4.7,10, 5,9.2, 5.1,8.8
2730 DATA 5,8.4, 4.5,8, 3.9,7.8, 3.9,8.7
2740 DATA 4.3,10.3, 4.9,11.4, 5.8,12.2, 1.7,8.2
2750 DATA 1.1,7.5, 0.9,6.7, 0.9,5.9, 1.2,4.2
2760 DATA 0.7,3.6, 0,3.6, 0,3.2, 1,3.3
2770 DATA 1.7,3.5, 2.5,3.6, 2.9,4.1, 2.9,4.7
2780 DATA 2.6,5.3, 2.1,6, 2,6.9, 2.7,7.8
2790 DATA 5,5.6, 4.3,4.9, 3.9,4, 3.7,3.2
2800 DATA 3.1,2.5, 2.2,2.1, 1.2,2.1, 0,2.4
2810 DATA 0,2, 1,1.7, 2,1.7, 3,2.1
2820 DATA 3.8,2.7, 4.5,3.5, 4.9,4.2, 5.1,5.1
2830 DATA 5.4,4.2, 5.1,3.5, 4.7,2.8, 4,2.2
2840 DATA 3.4,1.7, 2.6,1.4, 1.6,1.3, 0.8,1.2
2850 DATA 0,1.2, 5.9,3.2, 6.3,2.5, 6.6,2.3
2860 DATA 7.3,1.4, 4.6,0, 5.7,0.8, 6.3,1.7
2870 DATA 6.7,2.7, 6.8,3.7, 7.1,4.3, 7.2,4.7
2880 DATA 7.8,5.7, 8.6,7.3, 8.6,7.9, 7.5,9.9
2890 DATA 5.5,1, 6.1,1.9, 6.5,3.7, 6.7,4.5
2900 DATA 8,6.9, 8.2,7.8, 7.9,8.5, 7,9.9
2910 DATA 6.9,10.5, 7.2,8.2, 7.7,8.1, 7.9,7.7
2920 DATA 7.4,7.9, 6.9,7.9, 9.3,7.5, 9.8,8
2930 DATA 10.4,8.3, 11.1,8.3, 11.6,8, 11.8,7.7
2940 DATA 11.1,7.9, 10.5,7.9, 9.9,7.8, 8.5,6.7
2950 DATA 8.6,6.2, 8.2,5.5, 8.1,5.3, 8.1,4.9
2960 DATA 8.3,4.7, 8.7,4.7, 8.5,5, 9,4.6
2970 DATA 9.2,5, 9.4,5.1, 9.6,5, 9.4,4.8
2980 DATA 9.8,5.3, 9.9,5, 9.6,4.5, 0,0
2990 DATA 18.7,0, 18.7,14, 0,14
3000 DATA f
3010 REM=== points du trace ===
3020 DATA 1,2,3,4,5,6,7,8,9,10,f
3030 DATA 11,12,13,14,15,16,f
3040 DATA 12,17,18,19,20,21,22,f
3050 DATA 23,24,21,f
3060 DATA 24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,f
3070 DATA 5,16,22,26,f
3080 DATA 30,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60
```

```
3090 DATA 61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,79,79,80,80,81,f
3100 DATA 82,83,84,85,86,87,88,89,90,91,f
3110 DATA 82,75,f
3120 DATA 92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,48,f
3130 DATA 110,111,112,113,114,115,116,117,118,119,120,121,122
3140 DATA 123,124,125,126,127,113,f
3150 DATA 123,55,f,113,110,69,f
3160 DATA 62,128,129,130,131,132,133,134,f
3170 DATA 135,136,137,138,139,140,141,142,143,144,61,f
3180 DATA 145,146,147,148,149,150,151,152,f
3190 DATA 153,154,155,156,157,158,159,160,f
3200 DATA 161,162,163,164,165,166,167,168,169,f
3210 DATA 54,145,160,161,170,171,f
3220 DATA 172,173,1,f
3230 DATA 174,175,176,177,178,179,180,181,182,183,184,44,38,105,114,113,66,f
3240 DATA 174,185,186,187,188,189,190,191,192,193,45,f
3250 DATA 50,194,195,196,197,198,51,f
3260 DATA 199,200,201,202,203,204,205,206,207,199,f
3270 DATA 208,209,210,f
3280 DATA 211,212,213,214,215,213,f
3290 DATA 216,217,218,219,220,217,f,221,222,223,f
3300 DATA 224,225,226,227,224
3310 DATA fin
3320 LOCATE 10,10:PRINT phon$(0);
```

CONCLUSION :

L'AVENIR DE L'INTELLIGENCE ARTIFICIELLE

On a déjà beaucoup parlé, et on parlera encore beaucoup de ce que les japonais ont appelé (non sans prétention pensent certains) la cinquième génération d'ordinateurs. Or les japonais ne sont pas particulièrement connus pour leurs divagations philosophiques, mais bien plutôt pour leur solide bon sens, leur pragmatisme et leur sens commercial. Alors, cette cinquième génération, entièrement tournée vers l'intelligence artificielle et ses applications n'est-elle pas pour l'occident une réponse aux tergiversations de ceux qui détiennent le pouvoir de décider des grands axes de la recherche ?

Les japonais ont compris, avant tout le monde, l'énorme enjeu économique, mais aussi social et culturel que représentent toutes les applications de l'intelligence artificielle. Face à ces efforts fabuleux consentis à l'échelle nationale d'un des pays les plus riches du monde, les européens ont amorcé à leur tour une timide riposte : le projet ESPRIT, qui couvre plusieurs domaines prometteurs de l'intelligence artificielle, en est un exemple. Et bien sûr, les grands constructeurs, IBM à leur tête, n'hésitent pas à investir dans la recherche fondamentale ou appliquée, en espérant tirer bénéfice de la prochaine mutation de société que provoquera le déferlement de l'intelligence artificielle dans le grand public.

Les bases de connaissances vont complètement transformer les techniques de stockage de l'information : elles demanderont moins de place, les accès seront plus rapides, les mises à jour plus aisées. Bien au-delà d'un simple changement d'échelle de la technique, cette évolution va permettre à tout le monde d'accéder à n'importe quelle information, ou presque. C'est un véritable bouleversement qui se prépare : l'information bon marché et illimitée !

Le dialogue homme-machine permettra de lever les dernières barrières entre le grand public et les ordinateurs. Il suffira à n'importe qui de spécifier, en langage naturel, son problème, pour que l'ordinateur lui réponde. Plus besoin de programmes compliqués ni d'intermédiaire pour les écrire (informaticiens, etc.).

Les robots, dotés d'intelligence artificielle, vont se révéler de plus en plus à même de remplacer l'homme dans des tâches de plus en plus complexes. Et, bouleversement supplémentaire, ces tâches concerneront quasiment tous les domaines d'activités de l'homme et plus seulement quelques activités répétitives.

Les systèmes experts prendront le relais des experts humains dans des domaines aussi variés que la médecine, l'enseignement, l'administration, etc. Cela n'ira pas sans poser des problèmes de déontologie aux experts humains : accepteront-ils de transmettre à une machine une connaissance acquise durant toute une vie ?

Une véritable révolution se prépare sous nos yeux, elle ne manque pas de soulever des questions de tous ordres. Philosophiquement d'abord : jusqu'où l'homme ira-t-il dans sa quête scientifique ? et surtout quelles seront les conséquences de ses succès prévisibles ? de ses échecs ? Questions d'ordre moral également : quel impact aura sur nos activités l'introduction de l'intelligence artificielle et des robots ? Comment le grand public réagira-t-il face aux nouveaux pouvoirs mis à sa disposition ? Et surtout comment opérer sans douleur la mutation difficile vers une société de loisir où le chômage ne serait plus une tare mais un droit ? Cet état de fait ne risquerait-il pas à terme de provoquer l'apathie, puis la fin de l'espèce humaine ?

Toutes ces questions, et bien d'autres sont posées mais non résolues. Il conviendra d'y apporter toutes les réponses avant la révolution promise, et non après. Mais, est-ce possible ? En tout état de cause, la mutation est inéluctable : qu'elle se passe le mieux possible. Et comprenons que l'enjeu n'est pas seulement technique ou informatique. C'est une véritable aventure qui s'ouvre devant nous : pour notre part, nous ne voudrions la manquer sous aucun prétexte !

INDEX

CONSEILS DE LECTURE

Pour approfondir vos connaissances en BASIC, mieux connaître le système des CPC 464, 664 et 6128, et découvrir d'autres domaines de l'intelligence artificielle, P.S.I. vous propose une palette d'ouvrages utiles.

Pour maîtriser le BASIC Amstrad

- Basic Amstrad 1 – Méthode pratiques – Jacques Boisgontier et Bruno Césard (Editions du P.S.I.)

  Pour ceux qui ont déjà pratiqué un Basic, voici un ouvrage de perfectionnement au Basic Amstrad. Un chapitre sur le CP/M 2.2 et le CP/M Plus donne les principales commandes systèmes.

- Basic Amstrad 2 – Programmes – Jacques Boisgontier (Editions du P.S.I.)

  Pour pratiquer le Basic Amstrad, cet ouvrage donne de nombreux programmes de gestion, d'éducation et de jeu où le rôle des fichiers est expliqué et largement commenté.

- Basic plus – 80 routines sur Amstrad – Michel Martin (Editions du P.S.I.)

  Pour pousser votre Amstrad au maximum de ses capacités : 80 routines de simulation d'instructions qui n'existent pas en Basic Amstrad.

Pour mieux connaître le système des CPC et du PCW 8256

- Clefs pour Amstrad 1 – système de base – Daniel Martin (Editions du P.S.I.)

  Mémento présentant synthétiquement le jeu d'instructions du Z80, les points d'entrée des routines système, les connecteurs et brochages, etc.
  Le livre de chevet du programmeur sur Amstrad.

- Clefs pour Amstrad 2 – système disque – Daniel Martin et Philippe Jadoul (Editions du P.S.I.)

  Ce deuxième tome consacré au système disque présente les points d'entrée des routines disque, les blocs de contrôle, la programmation et les brochages des circuits spécialisés...
  La deuxième partie du livre est aussi destinée aux possesseurs d'Amstrad 8256.

- CP/M plus sur Amstrad – Yvon Dargery (Editions du P.S.I.)

  Toutes les commandes CP/M et CP/M plus pour maîtriser le système des 6128 et 8256 : un ouvrage de référence illustré par de nombreux programmes.

- Le livre de l'Amstrad – Tome 1 – Daniel Martin et Philippe Jadoul (BCM – diffusé par P.S.I.)

  Ce livre, destiné aux programmeurs des CPC 464 et 664, donne une étude complète de tous les circuits internes, et analyse la structure interne du BASIC. Vous y trouverez, en outre, une étude complète des RSX, et des programmes de scrolling, de traçage de rectangles, de coloriage de surface et de manipulation vectorielle.

Pour découvrir d'autres domaines de l'intelligence artificielle

- Trois étapes vers l'intelligence artificielle sur Amstrad CPC – René Descamps

- Trois étapes : apprentissage, jeux de réflexion et systèmes experts, sont illustrées par 20 programmes Basic. Chaque partie est abondamment commentée de façon à mettre l'intelligence artificielle à la portée de tous.

- Apple, logique et systèmes experts – René Descamps (Editions du P.S.I.)

  Une excellente introduction à la logique, illustrée par des programmes en Basic Applesoft accessibles à tous. Trois micros systèmes experts permettent de concevoir ce qu'il est possible de réaliser dans le domaine professionnel sur de gros ordinateurs.

- Systèmes industriels d'intelligence artificielle – Lucas Pun (Editests, diffusé par P.S.I.)

  Cet ouvrage s'adresse aux ingénieurs industriels, automaticiens et informaticiens qui ont à concevoir et à réaliser des Systèmes Industriels d'Intelligence Artificielle (SIIA).

Pour être informé régulièrement de l'actualité des micros Amstrad

□ MICROSTRAD, revue mensuelle du Groupe Tests.

Pour exploiter au mieux les capacités de votre micro, vous trouverez au sommaire de chaque numéro, un rendez-vous avec les rubriques clés :

- Découvrez la face cachée de votre CPC ou CPW : astuces, idées, conseils, tout pour comprendre votre micro, son anatomie, son fonctionnement, sa programmation et exploiter ses capacités graphiques et sonores.

- Domptez votre CPC 464, 664, 6128, PCW 8256 ou 8512 : passionnés, petits ou grands, spécialistes ou débutants, une information pratique et la compétence d'experts au service de votre micro.

- Programmez votre micro Amstrad : dans chaque numéro de MICROSTRAD, un cocktail de programmes (dessins, jeux, utilitaires, gestion, etc.) et des trucs de programmation.

I.A. SUR AMSTRAD CPC

Votre avis nous intéresse

Pour nous permettre de faire de meilleurs livres, adressez-nous vos critiques sur le présent ouvrage.

— Ce livre vous donne-t-il toute satisfaction ?

....

....

....

....

....

— Y a-t-il un aspect du problème que vous auriez aimé voir abordé ?

....

....

....

....

....

Si vous souhaitez des éclaircissements techniques, écrivez-nous, nous ne manquerons pas de vous répondre directement.

Où avez-vous acheté ce livre ?

☐ cadeau ☐ librairie ☐ autres
☐ exposition ☐ boutique micro

Comment en avez-vous eu connaissance ?

☐ publicité ☐ catalogue ☐ autres
☐ exposition ☐ conseils d'un ami

Avez-vous déjà acquis des livres P.S.I. ?

Lesquels ? ....

....

qu'en pensez-vous ? ....

....

....

Nom .... Prénom .... Age........
Adresse ....
Profession ....
Centre d'intérêt ....

A Frédéric
Merci pour le boulot sur
Amstrad CPC mémoire écrite
Thierry

CATALOGUE GRATUIT

Vous pouvez obtenir un catalogue complet des ouvrages PSI, sur simple demande, ou en retournant cette page remplie à votre libraire, à votre boutique micro ou aux

Editions du PSI
BP 86
77402 Lagny-sur-Marne Cedex

I.A. SUR AMSTRAD CPC LANGAGE ET FORMES

Concevoir des programmes en Basic d'Intelligence artificielle, c'est loin d'être impossible, et c'est à votre portée !

Vous qui **connaissez bien le Basic** Amstrad, vous allez explorer les dix logiciels commentés de ce livre pratique, et faire appel à toutes les ressources graphiques de votre Amstrad CPC.

Un robot déplaçant des cubes et des ballons illustre la *reconnaissance de formes*, et votre ordinateur, doué de "raison" vous surprend par ses dialogues en *langage naturel* et... son humour !

"Sophie" n'a aucun mal à prononcer devant vous les mots ou les phrases que vous lui soumettez. Enfin, un mini système expert diagnostique vos maladies imaginaires ou réelles, et vous prouve qu'il vous est possible de donner intelligence à votre machine.



ÉDITIONS DU P.S.I.
BP 86 - 77402 LAGNY S/MARNE CEDEX - FRANCE

ISBN 2-86595-325-4

ILLUSTRATION PASCALE COLLANGE

I.A. SUR AMSTRAD CPC - LANGAGE ET FORMES



AMSTRAD CPC



MÉMOIRE ÉCRITE
MEMORY ENGRAVED
MEMORIA ESCRITA



https://acpc.me/